THE UNIVERSITY
of EDINBURGH

Automatic Movie Analysis and Summarisation

Philip John Gorinski

Doctor of Philosophy
Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh
2017

Abstract

Automatic movie analysis is the task of employing Machine Learning methods to the field of screenplays, movie scripts, and motion pictures to facilitate or enable various tasks throughout the entirety of a movie's life-cycle. From helping with making informed decisions about a new movie script with respect to aspects such as its originality, similarity to other movies, or even commercial viability, all the way to offering consumers new and interesting ways of viewing the final movie, many stages in the life-cycle of a movie stand to benefit from Machine Learning techniques that promise to reduce human effort, time, or both. Within this field of automatic movie analysis, this thesis addresses the task of summarising the content of screenplays, enabling users at any stage to gain a broad understanding of a movie from greatly reduced data. The contributions of this thesis are four-fold: (i) We introduce ScriptBase, a new large-scale data set of original movie scripts, annotated with additional meta-information such as genre and plot tags, cast information, and log- and tag-lines. To our knowledge, ScriptBase is the largest data set of its kind, containing scripts and information for almost 1,000 Hollywood movies. (ii) We present a dynamic summarisation model for the screenplay domain, which allows for extraction of highly informative and important scenes from movie scripts. The extracted summaries allow for the content of the original script to stay largely intact and provide the user with its important parts, while greatly reducing the script-reading time. (iii) We extend our summarisation model to capture additional modalities beyond the screenplay text. The model is rendered multi-modal by introducing visual information obtained from the actual movie and by extracting scenes from the movie, allowing users to generate visual summaries of motion pictures. (iv) We devise a novel end-to-end neural network model for generating natural language screenplay overviews. This model enables the user to generate short descriptive and informative texts that capture certain aspects of a movie script, such as its genres, approximate content, or style, allowing them to gain a fast, high-level understanding of the screenplay. Multiple automatic and human evaluations were carried out to assess the performance of our models, demonstrating that they are well-suited for the tasks set out in this thesis, outperforming strong baselines. Furthermore, the ScriptBase data set has started to gain traction, and is currently used by a number of other researchers in the field to tackle various tasks relating to screenplays and their analysis.

Lay Summary

Summarisation is the task of producing a shorter version of a long medium, such as texts or movies, for example in order to enable people to understand its content in less time than would be required to consume the full version. Summarisation styles can be broadly divided into two categories: *Extractive* summarisation, which takes whole parts from the original input and presents those unchanged, and *abstractive* summarisation, which produces a completely new output for a given input. While summarising any given medium is usually a rather straight-forward task for humans, it is not so trivial to automate the process using computer programs. In this thesis, we show how we can generate *extractive* summaries for movie scripts as well as feature films and how we take a first step toward *abstractive* movie script summarisation by automatically producing short movie overview texts.

The first part of this thesis introduces a new corpus of movie scripts. Movie scripts are texts similar to theatre plays. They contain descriptions of what the camera sees, who the characters are, and what they are saying and doing. As such, movie scripts provide the basis for shooting a movie. We automatically search the internet for scripts and download them. In addition, we automatically obtain data about the movies from Wikipedia, IMDB – a large database containing various information such as a movie's production year, budget, cast and so on – as well as from Jinni, a movie recommendation website.

In the second part, we present a computer model that is able to automatically read a movie script and extract the most important scenes from it. We evaluate the model automatically, on pre-defined summaries, as well as manually, by having humans read and judge the summaries. Our model performs well; however, it takes only the movie script text into account.

In the third part of the thesis, we extend the model to work on script text as well as video. For this, we obtain the actual movies and feed them to a revised summarisation program. From the script text and the video, our model is able to generate a short

video, containing the most important segments of the movie. As before, we evaluate the generated summaries automatically and manually and show how the system using both text and video information performs better than systems using only the one or the other.

The final part of this thesis shows how we develop a computer model that can automatically analyse a movie script with respect to a variety of aspects, such as the movie's genre, general plot points, or content flags. The model classifies the scripts given these aspects and produces texts which accurately describe the content of movies.

Acknowledgements

First and foremost I would like to thank my principal supervisor Mirella Lapata, for supporting me throughout my PhD. The constant guidance, input, and knowledge she made available to me went beyond anything I could have imagined before starting to work with her. Every aspect of this thesis, and my academic work in general, was improved by her feedback.

I also want to thank my second supervisor Rik Sarkar, who provided me with a lot of help especially in the early stages of my work and throughout the years.

Thanks to my examiners, Shay Cohen and Dragomir Radev. I greatly appreciate the time they took to read my thesis, the stimulating discussion during my viva, and the valuable comments and corrections they provided.

Having been part of the School of Informatics in general, and ILCC in particular, was an academic and personal experience I consider myself more than lucky to have had. The whole group provided an outstanding environment not only for conducting my own research, but also for gaining insight into so many interesting aspects of other researchers' work, as well as access to an immense pool of knowledge. I want to especially thank Timothy Hospedales and Anestis Papazoglou, who provided useful information when I started to address Computer Vision in my work. Further thanks go to Stefanos Angelidis and Akash Srivastava for kick-starting my neural networks.

I could have never made it to the end of my PhD without the support of a great number of friends outside my academic environment. A special shout-out to the Lacrosse Bois, who are a force on and off the field.

Last but not least, I want to thank my parents Leonore and Jürgen without whose constant support in all things nothing would have been possible, and Tim, my favourite brother.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Philip John Gorinski*)

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Automatic movie analysis is the task of employing Machine Learning methods to screenplays, movie scripts[1] and motion pictures to facilitate and enable various tasks throughout the life-cycle of a movie. Some tasks, such as content flag identification to label a movie as child friendly or containing scenes of violence or nudity, can potentially be (almost) fully automated, while others such as script evaluation need to be human-centred, but may still benefit from computer-assisted approaches. Within the field of automatic movie analysis, this thesis addresses the challenges of movie summarisation under various aspects, utilising current Natural Language Processing, Video Processing and Machine Learning technology to summarise the content of movie scripts and feature films, enabling users at any stage to gain a broad understanding of a movie from greatly reduced data.

Such automatic processing methods promise to provide valuable help for dealing with the vast amount of data that is present at any given time during the life-cycle of a modern day feature film. For example, each year, about 50,000 screenplays are registered with the WGA[2], the Writers Guild of America. Only a fraction of these make it through to be considered for production, and an even smaller fraction make it to the big screen. How do producers and directors navigate through this vast number of scripts available? Typically, production companies, agencies, and studios hire script

[1] In technical terms, *movie scripts* are a type of *screenplay*, specifically written for a full feature film. In general, the term *screenplay* does refer to any work a screenwriter writes for a film, TV show, or even video game. Since we are expressly concerned with *movie* analysis in this work, we will use the terms *screenplay* and *(movie) script* interchangeably for the remainder of this thesis.

[2] The WGA is a collective term representing US TV and film writers.

readers, whose job is to analyse screenplays that come in, sorting the hopeful from the hopeless. Having read the script, a reader will generate a coverage report consisting of a logline (one or two sentences describing the story in a nutshell), a synopsis (a two- to three-page long summary of the script), comments explaining its appeal or problematic aspects, and a final verdict as to whether the script merits further consideration. Once the final version has been produced and wrapped, there are more challenges ahead. About 700 movies are released in the theatre each year[3]. An even greater number of films sees publication for TV, home cinema, or online platforms such as Netflix[4] or Hulu[5].

This large quantity of films make the domain attractive for a variety of automated or semi-automated (computer assisted) tasks. For example, in order to make a particular film attractive to the audience, studios typically release teasers and trailers, very short clips of not more than 180 seconds, designed to advertise the movie. In other scenarios such as multi-part movie releases, with individual film openings often split across several years, studios, cinemas or other providers might want to offer full-fledged summaries of previous parts, to remind the viewer of "what happened so far", and lead into the new release. Similarly, on-demand film platforms might make use of film summarisation techniques to either provide short summaries that users can watch before committing to a full movie, or they might even offer their users a more dynamic viewing experience, for example by letting them skim through films with focus on certain aspects ("show me all action scenes").

Beyond these challenges in the pre- and post-production process of a movie lie a variety of further tasks in the summarisation domain. Brief, high-level overviews of the content of a movie can be addressed as an abstractive summarisation task, in which certain aspects of a film are reflected in short natural language texts that describe the movie to a user. Examples of such short summaries are DVD cover texts describing the main plot, or info-texts on movie recommendation sites that inform readers about such aspects as genres, plot points and relevant content flags.

In this thesis we take a first step towards analysing and summarising the content of movies. We address three summarisation tasks which are relevant for different stages of the movie production cycle. In addition, we explore the synergies of text (movies start as screenplay) and video to develop models which are able to utilise informa-

[3] according to `http://www.boxofficemojo.com/`, 736 movies have been released in 2016

[4] `www.netflix.com`

[5] `www.hulu.com`

tion from both modalities in order to analyse and summarise movies. Working in the domains of screenplays and movies, we are also testing the maturity and robustness of current Natural Language Processing tools and technology, which are often geared towards quite different domains such as monologue or newspaper texts.

1.2 Literature and Film Analysis

For as long as there has been any form of media, there have been theories and endeavours pertaining to their analysis. With works of dramatic theory as old as Aristotle's "Poetics", the lineage and varieties of literary theory and criticism is too long and rich to discuss. The underlying principles of any given theory, however, are to evaluate and interpret works of art. In addition, the body of theory that is non-descriptive but aims at regularising and standardising their respective medium, is equally vast. As far back as the 1960s (Mosteller and Wallace, 1964), researchers, theorists and other interested groups have also tried to make use of mathematical (and later computational) models as part of analytical efforts.

In recent years, there has been a growing body of research that addresses the task of computer-aided analysis of media such as literature and film. Of particular interest in the analysis of *works of fiction* in these media is the notion of *characters*, as the protagonists of stories, i.e., the entities around who everything revolves. This point of view is taken by research that analyses books with respect to social networks between their characters (Elson et al., 2010), creates emotional trajectories for dramatic plays (Nalisnick and Baird, 2013) that capture characters' relations, or infers the archetypes of characters in summaries of books and films (Bamman et al., 2013, 2014). Similarly, in the visual domain, the notion of characters and networks based on them has been successfully employed to the field film analysis (Weng et al., 2009). The central approach in all cases is a graph-theoretic one, with the underlying assumption that certain aspects of the art work are inferable from the social structure of its characters.

In this thesis, we will also adopt a graph-theoretic perspective of movie analysis. Following the film-theoretic position that movies are "[about] nothing but character" (Monaco, 1982), we focus on the structural relations between the characters of a movie. We induce networks for screenplays and movies to model character-character interactions, and derive features from them that inform the summarisation process.

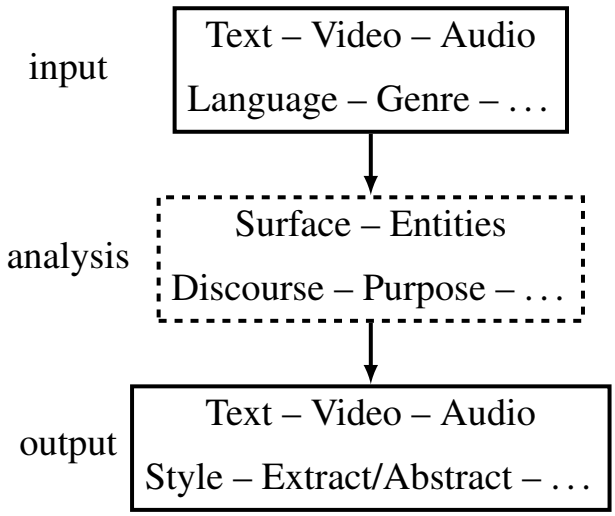


Figure 1.1: General outline of summarisation systems, and factors that can be used to categorise them as proposed by Mani and Maybury (1999); Jones et al. (1999). On the input level, the modalities (text, video, audio etc.) as well as linguistic factors like the source language or its genre can be taken into account. Systems can vary in the way they analyse the input, for example whether they make use of surface features like word counts, identify and correlate entities (sentences, paragraphs, objects), or even whether they are driven by a specific purpose of the created summary. Finally, variations on the output level include the output modalities, but also the style (words, sentences, pictures) of the summary, as well as whether the summary is generated by *extraction* of parts of the original source, or whether the system *abstracts* away from the input and generates entirely different outputs.

1.3 Summarisation

Summarisation is the task of reducing or compressing a large *source* into a quantitatively smaller *target* representation, while preserving as much of the original information as possible. In general, any summarisation system can be characterised with respect to its *input* modality, e.g., text, video, speech etc., the way it *analyses* the input, and the *output* that is being generated (see Figure 1.1). Some formalisations of these factors that characterise summarisation systems have been proposed, for example in Mani and Maybury (1999) or Jones et al. (1999). Typically, any given system is not strictly using only one approach, but a combination of features on any processing level.

Original efforts in automatic text summarisation concerned with the "Automatic Creation of Literature Abstracts" date all the way back to the 1950s (Luhn, 1958), and have seen much development over the decades. Many different approaches have been

established, from *statistical methods* using information gain (Mori, 2002) or other *tf-idf* based approaches (Neto et al. 2000; Lloret and Palomar 2009), to summarisation via *topic-modelling* and identification (Hearst 1997; Lin and Hovy 2000), and approaches analysing the graphical structure of elements contained in the source documents (Erkan and Radev, 2004). These approaches, while addressing the summarisation task from different angles, all have in common that they generate *extractive summaries*, i.e., the resulting output consists of words, sentences, or larger paragraphs as they are found in the source document. A different possibility is to generate *abstractive* summaries, which consist of content that is not present in the source. This line of research includes system employing sentence fusion to recombine original source sentences (Barzilay and McKeown, 2005), or more recently sentence compression using neural networks (Chopra et al., 2016).

Summarisation has an equally rich tradition outside the Natural Language Processing community, with video summarisation (also often called "video abstraction") research picking up with the advent of widely available (digital and analogue) video in the early 1990s (Otsuji et al. 1991; Teodosio and Bender 1993). As with texts, a large variety of approaches to summarising videos has been explored, from simple key-frame extraction based on shot boundaries (Smoliar and Zhang, 1994), or motion analysis (Wolf 1996; Ju et al. 1998), to recent *salience based* methods that identify objects, faces, etc. in video, and aim at a more coherent summarisation of the source by modelling the user's attention (Ma et al. 2002, 2005).

Naturally, multi-modal summarisation is a setting that follows these (and other) traditions by combining, for example, parallel text and video to enhance the summarisation quality (Ren et al. 2010; Evangelopoulos et al. 2013). This line of research is especially focused on the movie/TV show domain, where video and parallel texts such as subtitles are often readily available. The goal in this domain is to extract important *scenes* or sometimes *shots* from a very long video, in order to generate a shorter version that covers the full story of the film. This task of *movie summarisation* is in some regards similar to, but quite distinct from, the task of *trailer generation* (Pfeiffer et al., 1996; Lienhart et al., 1997; Smeaton et al., 2006). While similar techniques can be employed in both, the objectives of both are orthogonal: While a *summary* aims to include *all important information* contained in the source, ideally enabling full understanding of the video based on the summary alone, a *trailer* must *avoid* revealing too much important information, as its main goal is to entice the viewer to see the full movie.

In this thesis, we address the summarisation task for screenplays and movies in two modalities, text and video, as well as their combination in a multi-modal system. We generate *extractive* summaries for both modalities, by extracting written scenes from movie scripts, and visual scenes from films. We focus on a graph-based entity-centric analysis to inform the summarisation process. We also take a step toward abstract movie summarisation, by analysing movie script features, categorising the film under various aspects, and organising these aspects into high-level descriptions of the movie using an end-to-end neural network approach.

1.4 Natural Language Generation

Natural Language Generation is the task of generating natural language text that corresponds to a specific input. The task has been an active area of research for decades, with early systems such as Goldberg et al. (1994) using entries from a weather database to generate natural language forecasts, or generating technical documentation from knowledge bases (Reiter et al., 1995). Template-based systems, for example for summarisation like in DeJong (1982), can also be seen as NLG systems, organising keywords into pre-defined summary-template sentences. Such approaches, known as *pipelined* Natural Language Generation, typically consist of carefully engineered modules for content planning (what should go into the generated text), sentence planning (how many sentences should be produced, and what they contain), and sentence generation (realising the planned sentences as grammatically and morphologically correct output) (Reiter and Dale, 1997). While such systems can produce very high quality output, the engineering effort involved in all modules, often with hand-written domain specific rules and selection preferences makes them hard to adapt to new tasks, requiring large parts to be re-engineered.

With the onset of "big data" and better Machine Learning tools, recent years have seen growing research into data-driven Natural Language Generation. Such systems overcome the previously mentioned drawbacks by automatically learning some of the modules from data (Duboue and McKeown 2002; Barzilay and Lapata 2005, 2006; Lu and Ng 2011), or even by jointly learning all three (Angeli et al. 2010; Konstas and Lapata 2012). Even more recently, advances in the trainability and applicability of recurrent neural networks have made joint models a viable option for use in data-driven Natural Language Generation, with impressive results in varied settings such as machine translation (Cho et al. 2014b; Sutskever et al. 2014), dialogue (Wen et al., 2015),

or even poetry generation (Zhang and Lapata, 2014). In the context of movie analysis, neural networks offer an avenue to address the task of *abstractive* summarisation, as language models can be conditioned on the source, and can simultaneously have a completely separate output vocabulary.

In this thesis, we employ Neural Network Natural Language Generation to the movie script domain. We devise a joint neural network model for the task of generating descriptive overviews for films, based on features derived from screenplays. The model combines a feed-forward neural network *encoder* that is used to identify attributes that are applicable to the movie, and a Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) *decoder* that organises the attributes into natural language sentences.

1.5 Thesis Contributions

In this thesis, we present four main contributions to the field of automatic movie analysis in general, and to the areas of automatic screenplay and movie summarisation in particular.

A New, Large Scale Movie Script Corpus. We introduce *ScriptBase*, a new large-scale corpus consisting of a collection of scripts for a wide range of screenplays. The scripts are accompanied by a range of meta-data from various sources, providing information about release dates, genres, cast and characters, and more. *ScriptBase* is available in two varieties, *ScriptBase*-α and *ScriptBase*-J. *ScriptBase*-α represents more than $1,200$ movie scripts, including some transcripts and subtitle texts, with meta-data from the Internet Movie Database and Wikipedia. *ScriptBase*-J contains over 900 manually checked scripts from the original release, with additional rich meta-data from Jinni, a large movie database and recommendation engine. In addition to the plain-text version of movie scripts, *ScriptBase*-J also contains processed XML versions, containing sentence parses, coreference annotations, and more for the full scripts.

Movie Script Summarisation as Graph-Based Scene Extraction. We introduce a new task, movie script summarisation, and a new model based on character-character networks for the task of scene extraction. We motivate the desirability of such systems in the context of movie preproduction, where automated methods can be of aid at various points during writing and editing, pitching, or eventually for making a de-

cision about producing a film. We view the summarisation task as a scene extraction problem, requiring the system to identify those scenes in the script that are most informative. We address this task with a dynamic programming approach, finding chains of scenes that satisfy a global objective function which simultaneously optimises the chain for scene-to-scene progression, scene diversity, and overall scene importance. The results are extractive summaries, containing a subset of the original script scenes, reducing the reading requirements while maintaining a high degree of information. We also demonstrate the viability of this character-centric, graph-theoretic approach through automatic and human evaluations, which clearly favours the system built on these principles over competitive baselines.

Movie Summarisation via Multi-modal Scene Extraction. Building on the scene extraction model of the previous chapter, we develop a movie summarisation system that extracts visual summaries for feature films. We first adapt the previous graph-based script extraction model to the film modality by redefining the summarisation objective, and show how we employ Computer Vision techniques to obtain features based on video. Such a system can be useful, for example, in settings where users need a brief recap of a movie, or where users want to see the most important parts of a movie without having to sit through the entire feature length. We address the movie summarisation task with a dynamic programming model derived from the script-based system, and extend the objective definitions to include feature sets derived from screenplay texts as well as the corresponding feature films. The results are short videos, correlating in length to a fraction of the runtime of the source material, while still providing the viewer with a lot of the important information contained within the full film.

Joint Neural Network Overview Generation. We introduce a novel, jointly trained encoder-decoder neural network model, consisting of feed-forward encoders and a sentence planning LSTM, for the task of automatically generating high-level overviews for films. The motivation for this approach is to provide a fast and reliable way of automatically informing potential viewers about certain aspects of a movie, such as its major plot points, genre, or potential content warnings. We propose a joint neural network architecture combining feed-forward classifiers for attribute identification, and an LSTM for content selection and sentence generation. The input to the system are features derived from full movie scripts, resulting in short and accurate descriptions

of the aspects of movies. We demonstrate the effectiveness of this neural network approach to the abstractive analysis of screenplays in evaluations that show how human readers clearly prefer overviews that are produced by our joint architecture over texts that were generated with competitive baseline systems.

1.6 Thesis Outline

In this chapter, we have introduced the task of *automatic movie analysis*, in particular with emphasis on *automatic movie summarisation*. We have discussed the areas this thesis touches upon, and have briefly presented our contributions to the field. The remainder of this thesis is structured as follows:

Chapter 2 presents our efforts to assemble the *ScriptBase* corpus. We discuss how the scripts contained in the corpus were collected, and how we obtained the meta-data and additional resources. We also describe how we manually corrected the movie scripts, and automatically post-processed them into a machine readable format annotated with rich syntactical and semantic information.

Chapter 3 introduces *SceneSum*, our graph-based summarisation model for movie scripts. We devise an objective function that takes into account the scene-to-scene progression and diversity, as well as individual scene importance. We show how scripts can be analysed as character-character networks, and how we obtain features based on these graphs. The induced features are then used in the extraction process, and summaries are generated that jointly maximise the objective function. In automatic evaluations as well as user studies, we show that *SceneSum* is able to extract summaries from movie scripts that are more informative than summaries generated by competitive baseline systems.

Chapter 4 adapts *SceneSum* to the film domain. Once a movie has been produced, there are many settings in which automatically generated visual summaries can provide benefits. We show how we can re-interpret the objectives of the original system in the new setting, and present how we obtain video-based features to use in the graph-based scene extraction of video summaries. We also combine both modalities in a multi-modal system. We find that *SceneSum* can be successfully adapted to this new setting of generating visual summaries from movies. In a user study we also show

that summaries produced by the multi-modal *SceneSum* model consistently outperform summaries that were generated by baseline systems.

Chapter 5 presents MORGAN, our **M**ovie **O**ve**R**view **G**ener**A**tio**N** model. MORGAN is a joint neural network model for the task of generating overviews, high-level descriptions of certain aspects of a movie like its genre and style. Overviews are useful in settings where users need to be provided with short, descriptive and informative text about a movie, such as on a movie recommendation site. We introduce an encoder-decoder neural network for natural language overview generation that is based on feed-forward classifiers and an LSTM sentence planner and generator, and show how it can be jointly trained to generate overviews for screenplays. Experiments how that the encoder and decoder are well suited for their respective tasks, and that the joint model generates high quality overviews that are preferred by human judges over a number of competitive baselines.

Chapter 6 Concludes this thesis, and discusses directions we wish to pursue in future research.

Chapter 2

ScriptBase: A Movie Script Corpus

Movie scripts are a literary genre similar to that of theatre plays, and represent the first stage in the production process of a feature film. Scripts give detailed descriptions of movies by containing sections for each of the *scenes* in terms of their settings, actions and activities that go on off- and on-screen. They also describe the characters of the movie, and all their dialogues and interactions with each other. They are a highly structured type of literature, and provide the basis for producing and shooting feature films. As such, movie scripts represent a valuable resource for Natural Language Processing and Machine Learning, with potential in both research, e.g., in areas like dialogue and interaction analysis, theme identification or topic modelling, as well as commercial settings, where NLP applications for script processing could help with tasks such as script writing and reading, evaluation, or content flagging.

As many NLP and ML approaches require large amounts of data for training, development and testing, the acquisition and annotation – with meta-data and other information – of movie script collections is a valuable contribution to the research community.

2.1 Introduction

In this chapter, we introduce *ScriptBase*, a new corpus consisting of screenplays enriched with movie meta information. For the tasks outlined in this thesis, there is need for a reasonably sized data set of movie scripts, preferably with additional meta information such as genre, actors and characters, production year, and so on. Movie scripts are the actual written material used during the shooting of a movie, and as such contain not only dialogues between characters, but also scene divisions, narrations (e.g., *how* someone should perform and action; what is happening), descriptions of the envi-

ronment, and non-dialogue sequences that describe the progress of the movie, and are therefore very similar to scripts of theatre plays.

As such, movie scripts constitute a full-fledged literary genre, with various regulations and conventions writer have to adhere to, which are reflected in the scripts themselves. An example of such a movie script can be seen in Figure 2.1, taken from the movie "The Silence of the Lambs". Note in particular that certain parts are marked as required by script writing standards, like a *scene heading* beginning with `"INT."`, or different indentations for speakers and dialogue. Other conventions exist as well, such as the "one page per minute" rule of thumb, though these may be regularly broken. Structural rules as the ones mentioned before, on the other hand, should be held up by the writer. This is even more important given the fact that a script is only the first building block in a long (and hopefully lucrative) business, eventually leading to a full film release.

There exist some collections of scripts in previous literature, which are similar to *ScriptBase*, the data set presented here. Agarwal et al. (2014a) collected 222 movie scripts checked for structural sanity, and use them to generate and analyse "social networks" of movie characters, an idea we will pick up again in Chapter 3. Most notably, Danescu-Niculescu-Mizil and Lee (2011) presented a data set similar to *ScriptBase*, including screenplays and meta-data for 617 movies, aimed at the study of coordination of linguistic styles in dialogue. *ScriptBase* includes most of the scripts found within these corpora, but expands on them significantly. With over $1,200$ scripts (see Section 2.2.1), the corpus presented here doubles the number of scripts available in the largest of the previously available collections (Danescu-Niculescu-Mizil and Lee, 2011). Additionally, the new corpus contains a larger quantity of meta-data associated with the scripts, as well as user-written summaries, Wikipedia plot sections, and other data pertaining to the films. A second version of *ScriptBase* (Section 2.2.2) excludes some scripts from the larger collection, but still offers over 900 screenplays that have been manually checked and enriched with an even larger amount of meta-data covering different aspects of the screenplays.

While *ScriptBase* was compiled with the task of movie script summarisation in mind, we believe it offers a valuable resource for research on a wide variety of topics. Even though so far it has been available only on request, a number of other researchers are already actively using the corpus, on very varied tasks such as research on theme rewriting (Koncel-Kedziorski et al., 2016) or the analysis of power-structure in films (Sap et al., 2017), and more.

```
We can't get a good glimpse of his face, but his body is plump, above average
height; he is in his mid 30's.  Together they easily lift the chair into the
truck.
                              MAN (O.S.)
                  Let's slide it up, you mind?
CUT TO:
INT. THE PANEL TRUCK - NIGHT
He climbs inside the truck, ducking under a small hand winch, and grabs the
chair.  She hesitates again, but climbs in after him.

                              MAN
                  Are you about a size 14?

                              CATHERINE
                  (surprised)
                  What?

Suddenly, in the shadowy dark, he clubs her over the back of her head with
his cast.
```

Figure 2.1: Excerpt from the script of "The Silence of the Lambs". The scene heading `INT. THE PANEL TRUCK - NIGHT` denotes that the action takes place inside the panel truck at night. Character cues (e.g., `MAN`, `CATHERINE`) preface the lines the actors speak. Action/description lines describe what the camera sees (e.g., `We can't get a good glimpse of his face, but his body...`).

2.2 Collecting the corpus

We collected a large quantity of film scripts from various sources and organised them into two versions of the *ScriptBase* corpus, according to the quality of the crawled scripts, as well as the meta-data available for the respective movies. In this section, we describe how we obtained the various screenplays and meta-data, and organised them into the two collections.

2.2.1 *ScriptBase-α*

The majority of scripts contained in *ScriptBase* was obtained by crawling the *Internet Movie Script Database*[1] (IMSDB), a website containing a large quantity of original movie scripts, subtitle files, and some user generated transcriptions of films. In addition, we also crawled various smaller sites which provided scripts not covered by IMSDB (all script sources are acknowledged in Appendix A). Some scripts available online are for movies that have never been produced, or for very small "low-budget" films which are unknown to a larger audience. To ensure the availability of additional data for the films covered by our collection, the initially retrieved scripts were cross-matched against Wikipedia[2] and IMDB[3]. Only scripts that were represented in both resources were kept; scripts that could not be matched were discarded. This initial crawl and filter resulted in a total of 1,276 movies for which both scripts and meta-data were available.

After the scripts were collected, they were associated with information obtained through IMDB and Wikipedia. In particular, we include Wikipedia's *plot* sections (see Figure 2.2 for an example), which offer relatively concise and precise crowd sourced and crowd corrected[4] summaries of the movies. From IMDB, we obtained meta-data about the films, including their *release year*, *genres*, *IMDB rating*, and *cast*. We also add IMDB's user-written *synopses* and *summaries*, as well as *log lines* and *tag lines*. Synopses represent extremely detailed summaries of everything happening in a film. They are typically much larger than plots found on Wikipedia and, unlike the latter, contain information pertaining not only to important plot points, but to small details as well. The summaries found on IMDB are, on the other hand, typically smaller than plot sections, and only provide a high-level description of the film's plot. Log lines and tag lines are both extremely short snippets of information, typically not longer than two sentences. Log lines are one-sentence descriptions of the major idea behind the movie. They are useful, for example, when pitching a movie, or to give an extremely short idea about the film to a potential viewer. Tag lines, on the other hand, are typically used in marketing tools such as film posters, and are meant to entice the potential viewership to watch the movie. Figure 2.3 gives examples of meta-data contained in *ScriptBase*,

[1] `http://www.imsdb.com`

[2] `https://en.wikipedia.org`

[3] `http://www.imdb.com/`

[4] By *crowd sourced* and *crowd corrected* in the context of Wikipedia, we mean that articles are written by volunteers, and constantly updated and corrected, eventually producing a good and stable version of the plot.

She returns to Lecter, who tells her that the man is linked to Buffalo Bill. He offers to profile Buffalo Bill on the condition that he be transferred away from Chilton, whom he detests.

When Buffalo Bill kidnaps a U.S. Senator's daughter, Catherine Martin, Crawford authorizes Starling to offer Lecter a fake deal promising a prison transfer if he provides information that helps find Buffalo Bill and rescue the abductee.
Instead, Lecter begins a game of quid pro quo with Starling, offering comprehensive clues and insights about Buffalo Bill if Starling will give him information about her own past, something she was advised not to do.

Figure 2.2: Example extract from a Wikipedia plot section, for the movie "The Silence of the Lambs". Plot sections essentially amount to summaries of the respective movies.

while Figure 2.4 shows examples of log lines, tag lines and IMDB user summaries.

We call this initial version of the corpus containing $1,276$ scripts, IMDB and Wikipedia information *ScriptBase*-α. The corpus consists of movies comprising 23 genres, each is on average accompanied by 3 user summaries, 3 loglines, and 3 taglines. It spans years from 1909–2013. Some corpus statistics for the most represented genres found in the corpus are shown in Table 2.1.

	# Movies	AvgLines	AvgScenes	AvgCharacters
Drama	665	4484.53	79.77	60.94
Thriller	451	4333.10	91.84	52.59
Comedy	378	4303.02	66.13	57.51
Action	288	4255.56	101.82	59.99

Table 2.1: *ScriptBase*-α corpus statistics. Movies can have multiple genres, thus numbers do not add up to 1,276. Average number of lines and scenes per movie counted on the scripts. Average number of characters taken from meta-data.

meta-tag	value	
year	1991	
imdb score	8.6	
meta score	84	
genre	Crime	
genre	Thriller	
genre	Drama	
keyword	FBI	
keyword	serial killer	
keyword	psychiatrist	
keyword	agent	
...	...	
cast	Jodie Foster	Clarice Starling
cast	Lawrence A. Bonney	FBI Instructor
cast	Kasi Lemmons	Ardelia Mapp
...	...	...

Figure 2.3: *ScriptBase* meta-data, for the movie "The Silence of the Lambs". IMDB scores are the ratings for movies on IMDB itself, scored out of 10. Meta scores are taken from metacritic.com, a critical review aggregation website, which assigns scores out of 100. Keywords are generic tags that apply to the movie.

2.2.2 *ScriptBase*-J

While *ScriptBase*-α provides a large scale collection of scripts and meta-data, it has a few potential drawbacks: Firstly, the automatic crawl of movie scripts did not take into account the *type* of "script" that was retrieved. While the majority of crawled files are indeed full shooting scripts (as shown in Figure 2.1), some of them represent mere *transcripts* of movies, i.e., likely user-written text resembling scripts in that they mark who says what, and briefly describe what is happening, but lack the rich information provided by an actual movie script. Some other files may represent first drafts, versions of scripts that have been outlined but not yet fully fleshed out. Figure 2.5 gives an example of such a transcript retrieved in the initial crawl. Additionally, some of the retrieved files merely represented subtitle files (though lacking timestamps) of the film's dialogue, without any additional script-like information at all.

Secondly, the formatting of actual scripts is inconsistent. As Agarwal et al. (2014a)

logline	A young F.B.I. cadet must confide in an incarcerated and manipulative killer to receive his help on catching another serial killer who skins his victims.
tagline	Prepare yourself for the most exciting, mesmerising and terrifying two hours of your life!
tagline	To enter the mind of a killer she must challenge the mind of a madman.
tagline	May The Silence Be Broken!!
summary	Young FBI agent Clarice Starling is assigned to help find a missing woman to save her from a psychopathic serial killer who skins his victims. Clarice attempts to gain a better insight into the twisted mind of the killer by talking to another psychopath Hannibal Lecter, who used to be a respected psychiatrist. FBI agent Jack Crawford believes that Lecter, who is also a very powerful and clever mind manipulator, has the answers to their questions and can help locate the killer. However, Clarice must first gain Lecter's confidence before the inmate will give away any information.

Figure 2.4: Examples of user log lines, tag lines, and user summaries, for the movie "The Silence of the Lambs". Note that "summaries" on IMDB often do not cover the full extend of the movie's plot, as is the case here.

note, the formatting of a movie script provides valuable information during processing. For example, in a clean script, line indentation is used to mark different types of information such as the scene heading, speaker names, or spoken dialogue. However, different authors (and different sites) use different indentations, making the formatting of scripts contained in the original *ScriptBase*-α crawl inconsistent in this regard. Additionally, while inconsistencies *across* scripts are relatively easy to overcome by using simple heuristics, a large number of scripts also show inconsistencies *within* themselves. This makes automatic processing of full scripts potentially a lot harder, as these have to be addressed on a case-by-case basis.

Thirdly, in addition to mere inconsistencies with respect to indentations, many scripts also exhibit a certain level of noise. This includes within-script variations such as broken lines, character encoding errors, or site-specific formatting mistakes stemming from script conversions, such as page breaks and page numbers in plain-text documents. While such noise could be dealt with in an application that uses the scripts,

```
SCENE 4 Jane is packing her things, finds an advertisement in  the newspaper
asking for a driver to drive to the west coast.

SCENE 5
JANE That was in '78.  I was supposed to be the new Aretha, but the old Aretha
was the new Aretha and I was neither one,  so I was something that no one had
ever seen before.

ROBIN
Sure.
```

Figure 2.5: Example of a movie *transcript*, for the movie "Boys on the Side". Notice how important actual *script* information is missing from this transcript version.

they do add a potential source of errors downstream.

Finally, while *ScriptBase*-α already includes a lot of meta-data, for certain areas of research such as structured prediction tasks or tag-based Natural Language Generation, which this thesis addresses in Chapter 5, a richer set of additional information pertaining, in particular, to different aspects of movies such as their *genres*, *settings*, or *content flags* would be highly desirable.

ScriptBase-J addresses these potential shortcomings of the initial movie script crawl. We *manually* went through the full set of scripts contained in *ScriptBase*-α, and resolved issues wherever possible as follows: (i) We exclude from the new data set all "scripts" that represent either transcripts, rough drafts, or subtitle files, leaving us with a collection of true film shooting scripts. (ii) We manually de-noised the remaining scripts. For each screenplay in *ScriptBase*-J, we ensured consistent indentations *within* the script by manually correcting indentation levels for scene headings, description lines, character cues, and dialogue lines. In the same process, we also removed other types of formatting errors, and any other peculiarities we encountered on a case-by-case basis. As manual correction of scripts is extremely time consuming, we only corrected those screenplays that showed at least a minimum level of consistency and maximum level noise to begin with, and discarded screenplays that were deemed incorrigible within a reasonable amount of time. (iii) In addition to the initial Wikipedia and IMDB matching, we furthermore cross-referenced the movies of the original corpus against Jinni[5], a large database and movie recommendation engine. Jinni provides

[5] www.jinni.com

	All 14 Attributes	Plot	Place	Genre	Mood	Style
# values	827	416	199	31	29	27
Avg. per movie	34.7	14.2	2.3	2.8	4.5	1.3
Minimum per movie	15	4	0	1	0	0
Maximum per movie	62	28	9	7	12	8

Table 2.2: Statistics for attributes available in *ScriptBase*-J, for all attributes and top 5 attributes with most values.

an additional 14 types of meta-information for movies, conveying information about different aspects of the films. In particular, it indexes movies based on attributes and their values, which greatly expands on the *genre* information originally obtained via IMDB, by providing attribute-value pairs for the following aspects: *Attitudes* – how the movie is presented in terms of pacing and tone; intended *Audience*; whether the movie is *Based on* other material such as a book or theatre play; content *Flags*; the movie's overall *Genres*; the kind of *Humor* exhibited; the *Look* of a movie; its conveyed *Mood*; where the movie takes *Place*; general *Plot* cues; what *Praise* the movie has received; its musical *Score*; its overall *Style* – e.g., whether it is an ensemble film, or nonlinear; the *Time/Period* in which the movie is set. It furthermore organises these attributes into natural language overviews, which we also obtained. Examples of attributes and movie overviews that were added to the corpus are shown in Figure 2.6.

We call this new corrected and extended version of the corpus *ScriptBase*-J. It contains a total of 917 scripts that went through the manual correction process and could be successfully matched from *ScriptBase*-α to the Jinni database. Each script in *ScriptBase*-J is accompanied by the same information contained in the original corpus, with added meta-data and overviews from Jinni. This greatly extends the information previously collected through IMDB, in particular by adding tags for aspects other than *genre*. In total, the new data covers 14 different aspects, with a sum of over 800 values. Statistics on the new meta-data are shown in Table 2.2.

Figures 2.7 and 2.8 show distributions of the number of scenes and number of sentences per movie in *ScriptBase*-J, respectively. We can see that the majority of movies contains between 50 and 150 scenes, and between 2,500 and 3,500 script sentences, across all movies (top part of figures 2.7 and 2.8). These distributions seem to hold true when filtering movies for *genres*, as exemplified when analysing movies belonging to the *action, comedy, drama*, and *thriller* genres (middle and bottom parts).

Attribute	Values
Mood	Suspenseful, Captivating, Tense, Scary
Plot	Serial Killer, Special Agents, Investigation, Mind Game, Psychopath, Crimes, Deadly, Law Enforcement, Mind and Soul, Rivalry
Genre	Crime, Thriller
Style	Strong Female Presence
Attitude	Serious, Realistic
Place	Maryland, USA, Virginia
Period	20th Century, 90s
Based on	Based on Book
Praise	Award Winner, Blockbuster, Critically Acclaimed, Oscar Winner, Modern Classic, Prestigious Awards
Flag	Brief Nudity, Sexual Content, Strong Violent Content

Overview

The Silence of the Lambs can be described as tense, captivating, and suspenseful.
The plot revolves around special agents, mind games, and a psychopath.
The main genres are thriller and crime.
In terms of style, The Silence of the Lambs stars a strong female character.
In approach, it is serious and realistic.
It is located in Maryland and Virginia.
The Silence of the Lambs takes place in the 1990s.
It is based on a book.
The movie has received attention for being a modern classic, an Oscar winner, and a blockbuster.
Note that The Silence of the Lambs involves brief nudity and sexual content.

Figure 2.6: Jinni attributes, their values, and the corresponding overview for “The Silence of the Lambs”.

2.3 Processing Movie Scripts

To further facilitate using the corpus for a variety of tasks, we additionally post-processed each script contained in *ScriptBase*-J. As stated earlier, we manually reg-

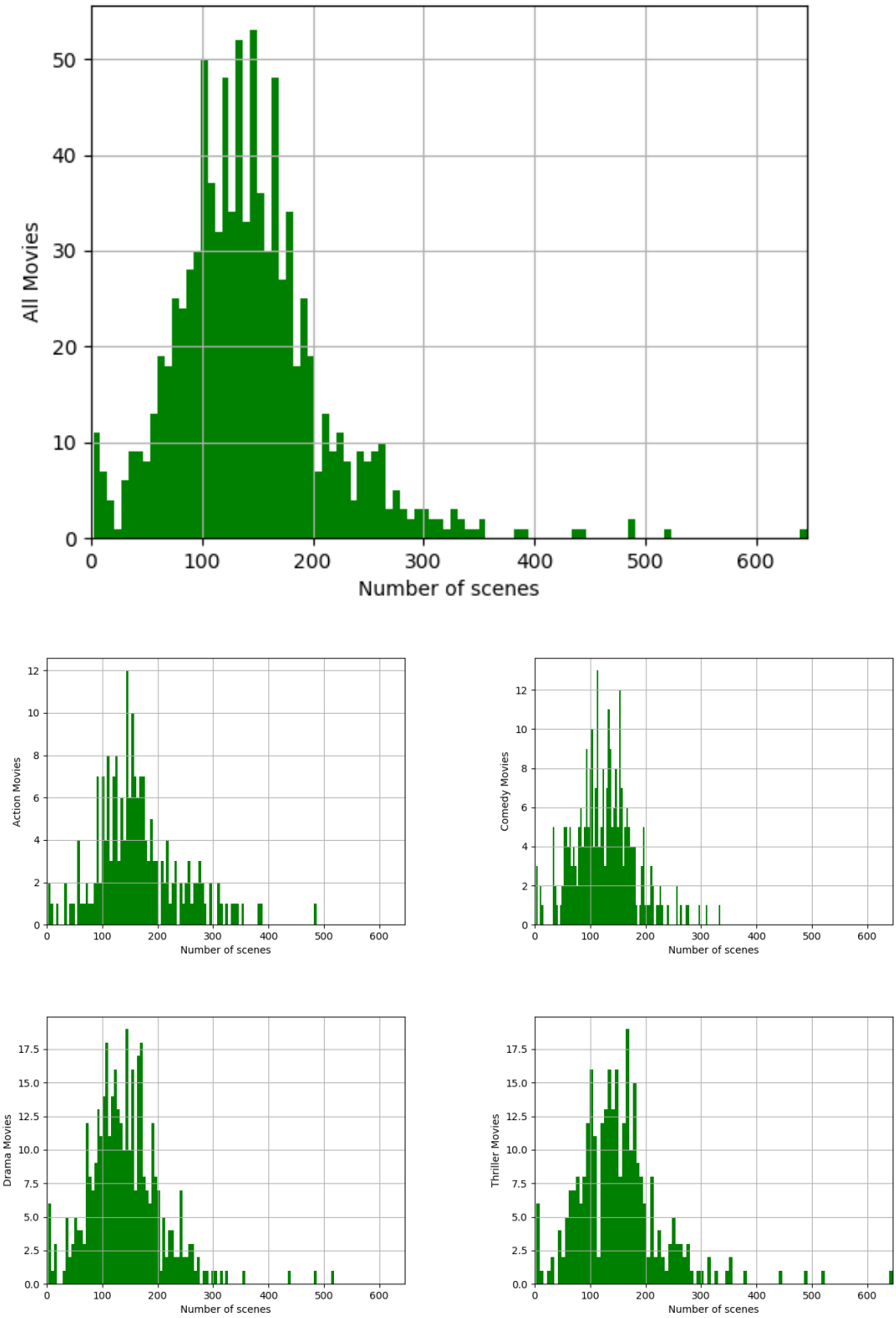


Figure 2.7: Distribution of number of scenes per movie, in *ScriptBase*-J. Top: All movies. Middle: Action/Comedy movies. Bottom: Drama/Thriller movies.

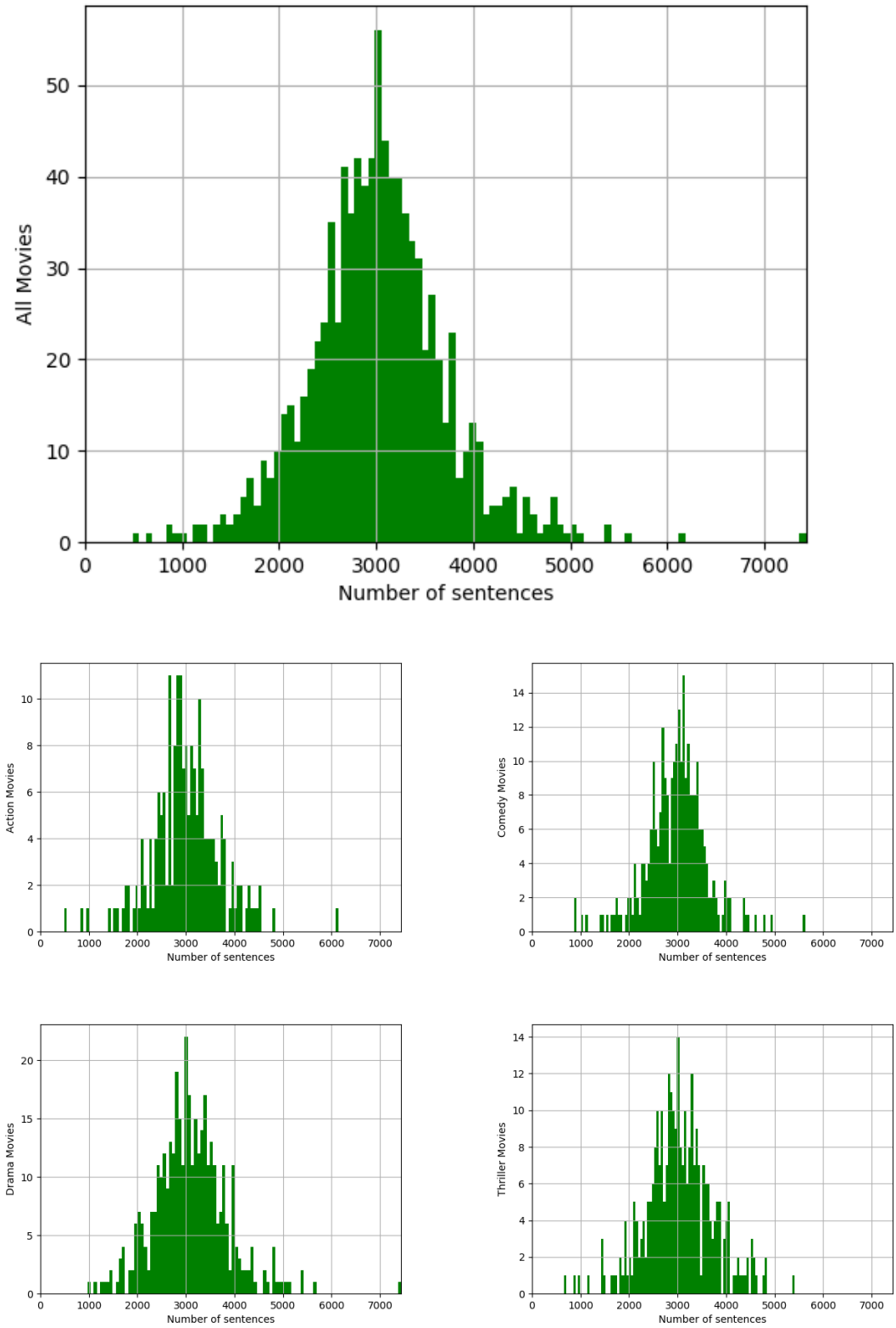


Figure 2.8: Distribution of number of sentences per movie, in *ScriptBase*-J. Top: All movies. Middle: Action/Comedy movies. Bottom: Drama/Thriller movies.

ularised indentations within each script. In a screenplay, similar to theatre plays, the indentation level of a line provides valuable information about what part of the screenplay it reflects (see also Figure 2.1). For example, *scene headings* typically start without any indentation, but contain a prefix like `INT. [...]` or `EXT. [...]`, describing where a scene takes place, i.e., in an *interior* or *exterior* location, respectively. Similarly, *descriptions* provide information about what the camera sees, or what can be heard. For dialogue parts, the *character cue* is typically indented the most, followed by the character's *speech*, which is less indented.

We organise this information into a machine-readable state, by converting all scripts into an XML format. We mark the different parts of a script and identify scene headings, descriptions, and so on, similar to Agarwal et al. (2014a). Furthermore, during conversion the scripts were annotated via the Stanford CoreNLP pipeline (Manning et al., 2014), to perform part-of-speech tagging, constituent parsing, named entity recognition and coreference resolution. They were also annotated with semantic roles (e.g., ARG0, ARG1), using the MATE tools (Björkelund et al., 2009). An example excerpt of a processed script is shown in Figure 2.9.

While most of these annotations could be used as generated by the pipeline, screenplays proved to pose some significant challenges during the coreference resolution step. We addressed these by introducing a number of post-processing heuristics, which allow us to correct certain types of mistakes, making use of the information intrinsically provided by the script. In particular, we employ the following heuristics after the initial coreference resolution step: (i) Using the *character cue* information contained within the script, we match personal pronouns in speech to specific chains. For first person pronouns, e.g., *I* or *myself*, we ensure that the coreference chain containing the pronoun is identified with the same chain that represents the current speaker. Similarly for second person pronouns, we identify their chain with that of the *previous* speaker, if available, under the assumption that the two characters are in conversation. (ii) We correct chains that refer to the same character or different characters, exploiting the fact that within the original script, each *character cue* identifies the unique character uttering the current text. This allows us to merge two chains that contain mentions of the same *character cue name*, as well as to separate chains that contain mentions of different cue names. (iii) We perform some smaller corrections, such as the same mention of a chain being covered by more than one sub-tree of a parse, and ensure the overall identified gender of a chain corresponds to the gender of its mentions.

The scripts contained in *ScriptBase*-J have been available on request, both as plain-

```
<scene count="18">
  <stageDirection count="0">INT. DR. LECTER'S CELL AND CORRIDOR - NIGHT (DIM LIGHT)</stageDirection>
  [...]
  <speech count="10" speaker="DR. LECTER" turn="3">
    <text>
      <sentence id="3" type="normal">
        <wordList>
          <word id="t_0" lemma="I" ne="O" pos="PRP" stem="i">I</word>
          <word id="t_1" lemma="do" ne="O" pos="VBD" stem="did">did</word>
          <word id="t_2" lemma="not" ne="O" pos="RB" stem="n't">n't</word>
          <word id="t_3" lemma="kill" ne="O" pos="VB" stem="kill">kill</word>
          <word id="t_4" lemma="he" ne="O" pos="PRP" stem="him">him</word>
          [...]
        </wordList>
        <graph>
          <nt head="t_1" id="nt_0" label="ROOT" span="t_0;t_10">
            <child id="nt_1"/>
          </nt>
          <nt head="t_1" id="nt_1" label="S" span="t_0;t_10">
            <child id="nt_2"/>
            <child id="nt_4"/>
            <child id="nt_21"/>
          </nt>
          <nt head="t_0" id="nt_2" label="NP" span="t_0;t_0">
            <child id="nt_3"/>
          </nt>
          <nt head="t_0" id="nt_3" label="PRP" span="t_0;t_0"/>
          [...]
        </graph>
        <semantics>
          <frame sense="kill.01" source="t_3">
            <role arg="A0" target="t_0"/>
            <role arg="AM-NEG" target="t_2"/>
            <role arg="A1" target="t_4"/>
          </frame>
          [...]
        </semantics>
      </sentence>
    </text>
  <speech>
  <coreference>
    [...]
    <chain gender="MALE" id="26" name="DR. LECTER">
      [...]
      <mention node="sc_18:sp_10:s_3:t_0" text="I"/>
      [...]
    </chain>
    <chain gender="UNKNOWN" id="113" name="">
      <mention node="sc_18:sp_10:s_3:t_4" text="him"/>
      [...]
  </chain>
  </coreference>
</scene>
```

Figure 2.9: Truncated example of processed XML for a part of the script of the movie "The Silence of the Lambs".

text and XML versions. With publication of this thesis, all scripts, meta-data and additional resources of *ScriptBase*-α, and *ScriptBase*-J, are also being made available through the University of Edinburgh Natural Language Processing Group's repository site[6].

[6] https://github.com/EdinburghNLP/scriptbase

2.4 Discussion

In this chapter, we have presented *ScriptBase*, a new large-scale corpus of movie scripts accompanied with rich meta-data and additional resources. We showed how the corpus was obtained, and how it was organised into two versions, *ScriptBase*-α and *ScriptBase*-J. The former collection contains scripts, transcripts and drafts for more than 1,200 automatically crawled movies, along with meta-data, synopses and user-written summaries from IMDB, as well as plot sections from Wikipedia. *ScriptBase*-J is built on top of *ScriptBase*-α, and contains over 900 manually corrected scripts along with a rich tag set of movie attributes and corresponding natural language overviews, in addition to the original IMDB and Wikipedia meta-data. We also showed how we processed scripts in *ScriptBase*-J to obtain machine readable XML versions for each screenplay, annotated with parses, coreference chains, and semantic roles retrieved via the application of state-of-the-art NLP tools. Each script in *ScriptBase*-J is also accompanied with the corresponding XML file. This new corpus promises to be a valuable resource in a large number of research areas in the domains of literary analysis, movie analysis, dialogue and interactions, and more. With the release of this thesis, the full collection has been made available to the research community, and as an on-request resource it is already being used in various research tasks.

For the future, we would like to transition more scripts from *ScriptBase*-α into *ScriptBase*-J. For this, the most restricting factor is the time it takes to manually correct script to ensure within-script consistency of indentations as well as de-noising. Another influence on the possible inclusion in the more feature-rich corpus is availability of meta-data that goes beyond the IMDB and Wikipedia sets, which is not guaranteed for all remaining scripts in *ScriptBase*-α. Finally, while we believe the processed XML versions of scripts contained in *ScriptBase*-J already offer a additional value, there is room for improvement in the processing stage, both with respect to the automatic coreference resolution, as well as the semantic role labelling. We would like to improve on those aspects in the future, employing either new and improved heuristics, new versions of the employed tools, or potentially even train and utilise specialised models for these tasks on the domain of movie scripts.

Chapter 3

Movie Script Summarisation as Graph-Based Scene Extraction

Automatic Movie Script Summarisation is the task of summarising the script of a feature film, using a Natural Language Processing approach. Movie scripts are large texts, often more than a hundred pages in length. At several points of the production process of a film, these scripts need to be read and understood, for example in order to decide whether they constitute a potentially successful movie, if they are similar to other already produced films, or what kind of story they convey. Summarising these texts offers a valuable help in these scenarios, greatly reducing reading time and, if done successfully, maintaining the key information of the original script, while only including the most necessary scenes. We show how this task of movie script summarisation can be addressed using a *character-centric* approach, as scene extraction based on character-character networks. The networks encode character interactions and relations, allowing for the assessment of scenes based on the properties of the network. We formalise the process of generating a shorter version of a screenplay as the task of finding an optimal chain of scenes, and develop a graph-based model that selects a chain by jointly optimising its logical progression, diversity, and importance. Human evaluation based on a question-answering task shows that our model produces summaries which are more informative compared to competitive baselines.

3.1 Introduction

Each year, about 50,000 screenplays are registered with the WGA[1], the Writers Guild of America. Only a fraction of these make it through to be considered for production and an even smaller fraction to the big screen. How do producers and directors navigate through this vast number of scripts available? Typically, production companies, agencies, and studios hire script readers, whose job is to analyse screenplays that come in, sorting the hopeful from the hopeless. Having read the script, a reader will generate a coverage report consisting of a logline (one or two sentences describing the story in a nutshell), a synopsis (a two- to three-page long summary of the script), comments explaining its appeal or problematic aspects, and a final verdict as to whether the script merits further consideration. A script excerpt from "Silence of the Lambs", an American thriller released in 1991, is shown in Figure 2.1, repeated here in Figure 3.1.

Although there are several screenwriting tools for authors (e.g., Final Draft is a popular application which automatically formats scripts to industry standards, keeps track of revisions, allows insertion of notes, and writing collaboratively online), there is a lack of any kind of script reading aids. Features of such a tool could be to automatically grade the quality of the script (e.g., thumbs up or down), generate synopses and loglines, identify main characters and their stories, or facilitate browsing (e.g., "show me every scene where there is a shooting").

In this chapter we explore whether current NLP technology can be used to address the task of script summarization, which we conceptualise as the process of generating a shorter version of a screenplay, ideally encapsulating its most informative scenes. The resulting summaries can be used to enhance script browsing, give readers a rough idea of the script's content and plotline, and speed up reading time.

So, what makes a good script summary? According to modern film theory, "all films are about nothing — nothing but character" (Monaco, 1982). Beyond characters, a summary should also highlight major scenes representative of the story and its progression. With this in mind, we define a script summary as a *chain of scenes* which conveys a narrative and smooth transitions from one scene to the next. At the same time, a good chain should incorporate some *diversity* (i.e., avoid redundancy), and focus on *important* scenes and characters. We formalise the problem of selecting a good summary chain using a graph-theoretic approach. We represent scripts as (directed) bipartite graphs with vertices corresponding to scenes and characters, and edge weights

[1]The WGA is a collective term representing US TV and film writers.

```
We can't get a good glimpse of his face, but his body is plump, above average
height; he is in his mid 30's.  Together they easily lift the chair into the
truck.
                              MAN (O.S.)
                  Let's slide it up, you mind?
CUT TO:
INT. THE PANEL TRUCK - NIGHT
He climbs inside the truck, ducking under a small hand winch, and grabs the
chair.  She hesitates again, but climbs in after him.

                              MAN
                  Are you about a size 14?

                              CATHERINE
                  (surprised)
                  What?

Suddenly, in the shadowy dark, he clubs her over the back of her head with
his cast.
```

Figure 3.1: Excerpt from the script of "The Silence of the Lambs". The scene heading `INT. THE PANEL TRUCK - NIGHT` denotes that the action takes place inside the panel truck at night. Character cues (e.g., `MAN`, `CATHERINE`) preface the lines the actors speak. Action/description lines describe what the camera sees (e.g., `We can't get a good glimpse of his face, but his body...`).

to their strength of correlation. Intuitively, if two scenes are connected, a random walk starting from one would reach the other frequently. We find a chain of highly connected scenes by jointly optimising logical progression, diversity, and importance.

The contributions covered in this chapter are three-fold: we introduce a novel summarization task, on a new text genre, and formalise scene selection as the problem of finding a chain that represents a film's story; we propose several novel methods for analysing script content (e.g., identifying important characters and their interactions); and perform a large-scale human evaluation study using a question-answering task. Experimental results show that our method produces summaries which are more informative compared to several competitive baselines.

3.2 Related Work

Computer-assisted analysis of literary text has a long history, with the first studies dating back to the 1960s (Mosteller and Wallace, 1964). More recently, the availability of large collections of digitised books and works of fiction has enabled researchers to observe cultural trends, address questions about language use and its evolution, study how individuals rise to and fall from fame, perform gender studies, and so on (Michel et al., 2010). Most existing work focuses on low-level analysis of word patterns, with a few notable exceptions. Elson et al. (2010) analyse 19th century British novels by constructing a conversational network with vertices corresponding to characters and weighted edges corresponding to the amount of conversational interaction. Elsner (2012) analyses characters and their emotional trajectories, whereas Nalisnick and Baird (2013) identify a character's enemies and allies in plays based on the sentiment of their utterances. Other work (Bamman et al., 2013, 2014) automatically infers latent character types (e.g., villains or heroes) in novels and movie plot summaries. Sang and Xu (2010a) present a first approach to summarise movies using attention analysis, based on character interactions.

Although we are not aware of any previous approaches to summarise screenplays, the field of Computer Vision is rife with attempts to summarise video (see Reed, 2004; Money and Agius, 2008 overviews). Most techniques are based on visual information and rely on low-level cues such as motion, colour, or audio (e.g., Rasheed et al. 2005). Movie summarization is a special type of video summarization which poses many challenges due to the large variety of film styles and genres. A few recent studies (Weng et al., 2009; Lin et al., 2013) have used concepts from social network analysis to identify lead roles and role communities in order to segment movies into scenes (containing one or more shots) and create more informative summaries. A surprising fact about this line of work is that it does not exploit the movie script in any way. Roles are identified using face recognition techniques and scene boundaries are presumed unknown and are automatically detected.

Our own approach is inspired by work in egocentric video analysis. An egocentric video offers a first-person view of the world and is captured from a wearable camera focusing on the user's activities, social interactions, and interests. Lu and Grauman (2013) present a summarization model that extracts subshot sequences while finding a balance of important subshots that are both diverse and provide a natural progression through the video, in terms of prominent visual objects (e.g., bottle, mug, television).

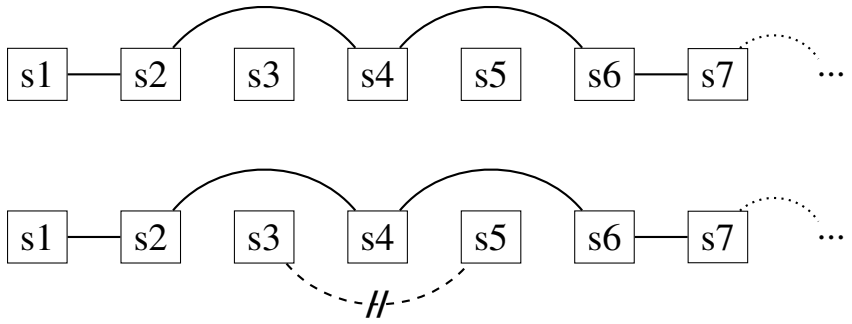


Figure 3.2: Example of consecutive chain (top). Squares represent scenes in a screenplay. The bottom chain would not be allowed, since the connection between $s3$ and $s5$ makes it non-consecutive.

We adapt their technique to our task, and show how to estimate character-scene correlations based on linguistic analysis. We also interpret movies as social networks and extract a rich set of features from character interactions and their sentiment which we use to guide the summarization process.

3.3 The Scene Extraction Model

As mentioned earlier, we define script summarization as the task of selecting a chain of scenes representing the movie's most important content. We interpret the term scene in the screenplay sense. A scene is a unit of action that takes place in one location at one time (see Figure 3.1). Within a movie script, these segments are typically marked, mostly via identifiers like `EXT` or `INT`, denoting scenes that take place inside or outside of some location, respectively. For the task at hand, we therefore need not be concerned with scene segmentation; scene boundaries are clearly marked, and constitute the basic units over which our model operates.

Let $M = (S,C)$ represent a screenplay consisting of an ordered set $S = (s_1, s_2, \ldots, s_n)$ of scenes, and an ordered set $C = (c_1, \ldots, c_m)$ of characters. We are interested in finding a sequence $S' = (s_i, \ldots, s_k); 1 \leq i,k \leq n$ of *ordered, consecutive* scenes subject to a compression rate m (see the example in Figure 3.2). A natural interpretation of m in our case is the percentage of scenes from the original script retained in the summary. The extracted chain should contain (a) *important* scenes (i.e., critical for comprehending the story and its development); (b) *diverse* scenes that cover different aspects of the movie's story; and (c) scenes that highlight the story's *progression* from beginning to end. We therefore find the chain S' maximising the objective function $Q(S')$ which is

the weighted sum of three terms: the story progression P, scene diversity D, and scene importance I:

$$S^* = \underset{S' \subset S}{arg\,max}\, Q(S') \tag{3.1}$$

$$Q(S') = \lambda_P P(S') + \lambda_D D(S') + \lambda_I I(S') \tag{3.2}$$

In the following, we define each of the three terms.

Scene-to-scene Progression The first term in the objective is responsible for selecting chains representing a logically coherent story. Intuitively, this means that if our chain includes a scene where a character commits an action, then scenes involving affected parties or follow-up actions should also be included. We operationalise this idea of progression in a story in terms of how strongly the characters in a selected scene s_i influence the transition to the next scene s_{i+1}:

$$P(S') = \sum_{i=0}^{|S'|-1} \sum_{c \in C_i} INF(s_i, s_{i+1}|c) \tag{3.3}$$

We represent screenplays as weighted, bipartite graphs connecting scenes and characters:

$$B = (V,E) \,:\, V = C \cup S$$

$$E = \{(s,c,w_{s,c})|s \in S,\, c \in C,\, w_{s,c} \in [0,1]\} \cup$$

$$\{(c,s,w_{c,s})|c \in C,\, s \in S,\, w_{c,s} \in [0,1]\}$$

The set of Vertices V corresponds to the union of characters C and scenes S. We therefore add to the bipartite graph one node per scene and one node per character, and two directed edges for each scene-character and character-scene pair. An example of such a bipartite graph is shown in Figure 3.3. We further assume that two scenes s_i and s_{i+1} are tightly connected in such a graph if a random walk with restart (RWR; Tong et al. (2006), Kim et al. (2014)) which starts in s_i has a high probability of ending in s_{i+1}.

In order to calculate the random walk stationary distributions, we must estimate the edge weights from characters to scenes, and from scenes to characters in the bipartite graph. Intuitively, we are interested in how important a character is generally in the movie, and specifically in a particular scene. For edge weights $w_{c,s}$ from *characters* to *scenes*, we consider the probability of a character being important in the movie overall, i.e., of them belonging to the set of main characters:

$$w_{c,s} = P(c \in main(M)),\ \forall (c,s,w_{c,s}) \in E \tag{3.4}$$

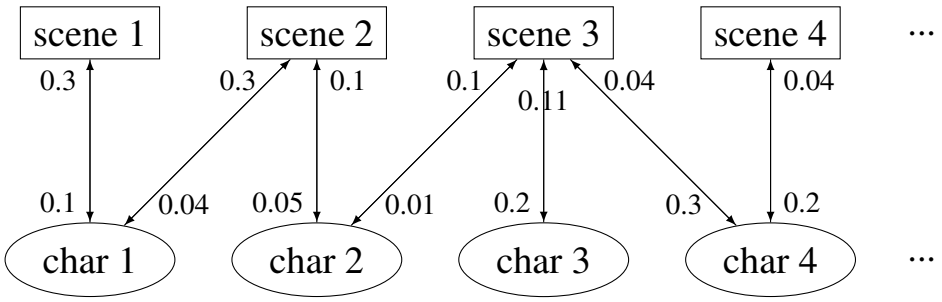


Figure 3.3: Example of a bipartite graph, connecting a movie's scenes with participating characters.

where $P(c \in main(M))$ is some probability score associated with c being a main character in script M. We use $P(c \in main(M))$ for all outgoing edges of character c, i.e., $w_{c,s}$ represent a prior over the movie's characters. This reflects our assumption that any given character retains the same likelihood of being a main character throughout the entirety of the movie, independent of the scene they are appearing in. In contrast to this, for the weights from scenes to characters $w_{c,s}$, we take the number of interactions a character is involved in within a specific scene relative to the total number of interactions in that scene, as indicative of the character's importance for this scene in isolation. Interactions refer to conversational interactions as well as relations between characters (e.g., who does what to whom):

$$w_{s,c} = \frac{\sum_{c' \in C_s} inter(c, c')}{\sum_{c_1, c_2 \in C_s} inter(c_1, c_2)}, \forall (s, c, w_{s,c}) \in E \tag{3.5}$$

We defer discussion of how we model the probability $P(c \in Main(M))$ and obtain interaction counts to Section 3.4. Finally, weights $w_{c,s}$ and $w_{s,c}$ in the bipartite graph B are normalised:

$$w_{c,s} = \frac{w_{c,s}}{\sum_{(c,s',w'_{c,s})} w'_{c,s}}, \forall (c, s, w_{c,s}) \in E \tag{3.6}$$

$$w_{s,c} = \frac{w_{s,c}}{\sum_{(s,c',w'_{s,c})} w'_{s,c}}, \forall (s, c, w_{s,c}) \in E \tag{3.7}$$

We read off the stationary distributions of a random walk from a transition matrix T, enumerating over all vertices v (i.e., characters *and* scenes) in the bipartite graph B:

$$T(i, j) = \begin{cases} w_{i,j} \text{ if } (v_i, v_j, w_{i,j}) \in E^B \\ 0 \text{ otherwise} \end{cases} \tag{3.8}$$

We measure the *influence* (*INF*, Equation (3.3)) that individual characters have on scene-to-scene transitions as follows. The stationary distribution r_k for a RWR walker starting at node k is a vector that satisfies:

$$r_k = (1-\varepsilon)Tr_k + \varepsilon e_k \tag{3.9}$$

where T is the transition matrix of the graph, e_k is a *seed vector*, with all elements 0, except for element k which is set to 1, and ε is a restart probability parameter. In practice, our vectors r_k and e_k are indexed by the scenes and characters in a movie, i.e., they have length $|S|+|C|$, and their n_{th} element corresponds either to a known scene or character. In cases where graphs are relatively small, we can compute r directly[2] by solving:

$$r_k = \varepsilon(I-(1-\varepsilon)T)^{-1}e_k \tag{3.10}$$

The lth element of r then equals the probability of the random walker being in state l in the stationary distribution. Let r_k^c be the same as r_k, but with the character node c of the bipartite graph being turned into a sink, i.e., all entries for c in the transition matrix T are 0. We can then define how a single character influences the transition between scenes s_i and s_{i+1} as:

$$INF(s_i, s_{i+1}|c) = r_{s_i}[s_{i+1}] - r^c_{s_i}[s_{i+1}] \tag{3.11}$$

where $r_{s_i}[s_{i+1}]$ is shorthand for that element in the vector r_{s_i} that corresponds to scene s_{i+1}. We use the *INF* score directly in Equation (3.3) to determine the progress score of a candidate chain.

Diversity The diversity term $D(S')$ in our objective should encourage chains which consist of more dissimilar scenes, thereby avoiding redundancy. We take the diversity of chain S' as the sum of the diversities of its successive scenes:

$$D(S') = \sum_{i=1}^{|S'|-1} d(s_i, s_{i+1}) \tag{3.12}$$

The diversity $d(s_i, s_{i+1})$ of two scenes s_i and s_{i+1} is estimated taking into account two factors: (a) do they have any characters in common, and (b) does the sentiment change from one scene to the next:

$$d(s_i, s_{i+1}) = \frac{d_{char}(s_i, s_{i+1}) + d_{sen}(s_i, s_{i+1})}{2} \tag{3.13}$$

[2]We could also solve for r iteratively which would be preferable for large graphs, since the performed matrix inversion is computationally expensive.

where $d_{char}(s_i, s_{i+1})$ and $d_{sen}(s_i, s_{i+1})$ denote the character and sentiment similarities between scenes. Specifically, $d_{char}(s_i, s_{i+1})$ is the relative character overlap between scenes s_i and s_{i+1}:

$$d_{char}(s_i, s_{i+1}) = 1 - \frac{|C_{s_i} \cap C_{s_{i+1}}|}{|C_{s_i} \cup C_{s_{i+1}}|} \tag{3.14}$$

d_{char} will be 0 if two scenes share the same characters and 1 if no characters are shared. Analogously, we define d_{sen}, the sentiment overlap between two scenes as:

$$d_{sen}(s_i, s_{i+1}) = 1 - \frac{k \cdot dif(s_i, s_{i+1})}{k - k \cdot dif(s_i, s_{i+1}) + 1} \tag{3.15}$$

$$dif(s_i, s_{i+1}) = \frac{1}{1 + |sen(s_i) - sen(s_{i+1})|} \tag{3.16}$$

where the sentiment $sen(s)$ of scene s is the aggregate sentiment score of all interactions in s:

$$sen(s) = \sum_{c,c' \in C_s} sen(inter(c, c')) \tag{3.17}$$

We explain how interactions and their sentiment are computed in section 3.4. Again, d_{sen} is larger if two scenes have a less similar sentiment. The term for sentiment difference $dif(s_i, s_{i+1})$ becomes 1 if the sentiments are identical, and increasingly smaller for more dissimilar sentiments. The sigmoid-like function in Equation (3.15) scales d_{sen} within range $[0, 1]$ to take smaller values for larger sentiment differences, where the factor k adjusts the curve's smoothness.

Importance The score $I(S')$ captures whether a chain contains overall important scenes. We define $I(S')$ as the sum of all scene-specific importance scores $imp(s_i)$ of scenes contained in the chain:

$$I(S') = \sum_{i=1}^{|S'|} imp(s_i) \tag{3.18}$$

The importance $imp(s_i)$ of a scene s_i is the ratio of lead to support characters within that scene:

$$imp(s_i) = \frac{\sum_{c:\, c \in C_{s_i} \wedge c \in main(M)} 1}{\sum_{c:\, c \in C_{s_i}} 1} \tag{3.19}$$

where C_{s_i} is the set of characters present in scene s_i, and $main(M)$ is the set of main characters in the movie.[3] $I(s_i)$ is 0 if a scene does not contain any main characters, and 1 if it contains only main characters (see Section 3.4 for how $main(M)$ is inferred).

[3] Whether scenes are important if they contain many main characters is an empirical question in its own right. For our purposes, we assume that this relation holds.

Optimal Chain Selection We use Linear Programming to efficiently find a good chain. The objective is to maximise Equation (3.2), i.e., the sum of the terms for progress, diversity and importance, subject to their weights λ. We add constraints for the number of scenes and enforce the linear order of the selected scenes by adding constraints that forbid non consecutive combinations. We use GLPK[4] to solve the linear problem.

3.4 Implementation

In this section we discuss several aspects of the implementation of the model presented in the previous section. We explain how interactions are extracted and how sentiment is calculated. We also present our method for identifying main characters and estimating the weights $w_{c,s}$ and $w_{s,c}$ in the bipartite graph. Our summarization experiments focused on comedies and thrillers, for which we randomly selected 30 movies for training/development and 66 movies for testing from *ScriptBase*-J (Chapter 2).

3.4.1 Interactions

The notion of an *interaction* underlies many aspects of the model defined in the previous section. For instance, interaction counts are required to estimate the weights $w_{s,c}$ in the bipartite graph of the progression term (see Equation (3.5)), and in defining diversity (see Equations (3.15)–(3.17)). As we shall see below, interactions are also important for identifying main characters in a screenplay.

We use the term interaction to refer to *conversations* between two characters, as well as their *relations* (e.g., if a character kills another). For conversational interactions, we simply need to identify the *speaker* generating an utterance and the *listener* at which the utterance is directed. Speaker attribution comes for free in our case, as speakers are clearly marked in the text (see Figure 3.1). Listener identification is more involved, especially when there are multiple characters in a scene. We rely on a few simple heuristics. We assume that the previous speaker in the same scene, who is different from the current speaker, is the listener. If there is no previous speaker, we assume that the listener is the closest character mentioned in the speaker's utterance (e.g., via a coreferring proper name or a pronoun). In cases where we cannot find a suitable listener, we assume the current speaker is the listener. We obtain character

[4]https://www.gnu.org/software/glpk/

relations from the output of a semantic role labeler (Björkelund et al., 2009). Relations are denoted by verbs whose ARG0 and ARG1 roles are character mentions or pronouns. We extract relations from the dialogue but also from scene descriptions. For example, in Figure 3.1 the description `Suddenly, [...] he clubs her over the head` contains the relation `clubs(MAN,CATHERINE)`. Pronouns are resolved to their antecedent using the Stanford coreference resolution system (Lee et al., 2011).

3.4.2 Sentiment

We labelled lexical items in screenplays with sentiment values using the AFINN-96 lexicon (Nielsen, 2011), which is essentially a list of words scored with sentiment strength within the range $[-5,+5]$. The list also contains obscene words (which are often used in movies) and some Internet slang. By summing over the sentiment scores of individual words, we can work out the sentiment of an interaction between two characters, the sentiment of a scene (see Equation (3.17)), and even the sentiment between characters overall (e.g., who likes or dislikes whom in the movie in general).

3.4.3 Main Characters

The progress term in our summarization objective crucially relies on characters and their importance (see the weight $w_{c,s}$ in Equation (3.4)). Previous work (Weng et al., 2009; Lin et al., 2013; Agarwal et al., 2014a) extracts social networks where nodes correspond to roles in the movie, and edges to their co-occurrence. Leading roles (and their communities) are then identified by measuring their centrality in the network (i.e., number of edges terminating in a given node).

It is relatively straightforward to obtain a social network from a screenplay. Formally, for each movie we define a *weighted* and *undirected* graph:

$$G = \{C,E\} \; : \; C = \{c_1, \ldots c_n\}, \; E = \{(c_i, c_j, w) | c_i, c_j \in C, \, w \in \mathbb{N}_{>0}\} \tag{3.20}$$

where vertices correspond to movie characters[5], and edges denote character-to-character interactions. Figure 3.4 shows an example of a social network for "The Silence of the Lambs". Due to lack of space, only main characters are displayed, however the actual graph contains *all* characters (42 in this case). Importantly, edge

[5]We assume one node per *speaking role* in the script.

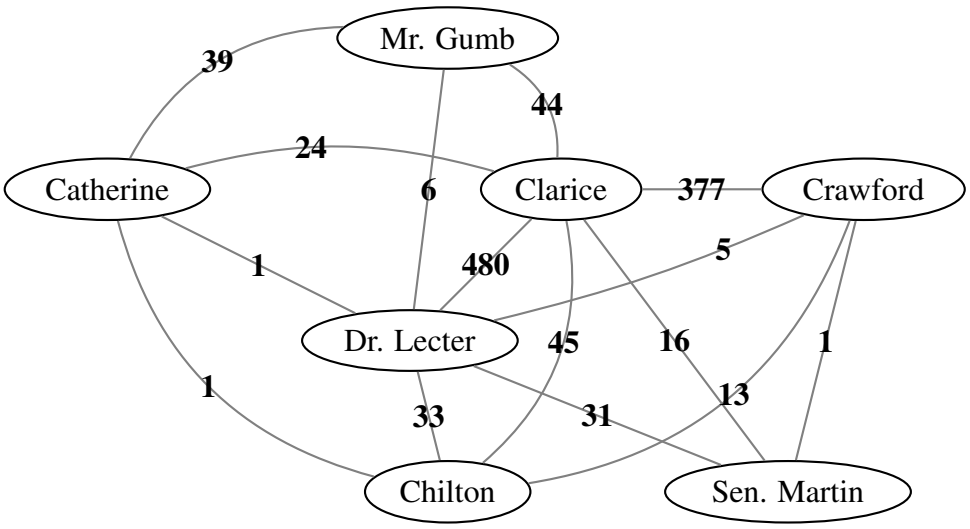


Figure 3.4: Social network for "The Silence of the Lambs"; edge weights correspond to absolute number of interactions between nodes.

weights are not normalised, but directly reflect the strength of association between different characters.

We do not solely rely on the social network to identify main characters. We estimate $P(c \in main(M))$, the probability of c being a leading character in movie M, using a Multi Layer Perceptron (MLP) and several features pertaining to the structure of the social network and the script text itself. A potential stumbling block in treating character identification as a classification task is obtaining training data, i.e., a list of main characters for each movie. We generate a gold-standard by matching the screenplay's characters against Wikipedia data, assuming that the characters listed under Wikipedia's *Cast* section (or an equivalent section, e.g., *Characters*) are the main characters in the respective movie.

Examples of the features we used for the main character classification task include the barycenter of a character (i.e., the sum of its distance to all other characters), PageRank (Page et al., 1999), an eigenvector-based centrality measure, absolute/relative interaction weight (the sum of all interactions a character is involved in, divided by the sum of all interactions in the network), absolute/relative number of sentences uttered by a character, number of times a character is described by other characters (e.g., `He is a monster` or `She is nice`), number of times a character talks about other characters, and type-token-ratio of sentences uttered by the character (i.e., rate of unique words in a character's speech). Using the described features, the MLP achieves an F1 of 79.0% on the test set. It outperforms other classification meth-

ods such as Naive Bayes or logistic regression. Using the full-feature set, the MLP also obtains performance superior to any individual measure of graph connectivity.

Aside from Equation (3.4), lead characters also appear in Equation (3.19), which determines scene importance. We assume a character $c \in main(M)$ as a main character of the movie if it is predicted by the MLP with a probability ≥ 0.5.

3.5 Experimental Setup

3.5.1 Gold Standard Chains

The development and tuning of the chain extraction model presented in Section 3.3 necessitates access to a gold standard of key scene chains representing the movie's most important content. As our experiments concentrate on a selection of 96 comedies and thrillers, performing the scene selection task for such a big corpus manually would be both time consuming and costly. Instead, we used distant supervision based on Wikipedia to automatically generate a gold standard.

Specifically, we assume that Wikipedia plots are representative of the most important content of a movie. Using an alignment algorithm similar to the algorithm presented in Nelken and Shieber (2006), we automatically align script sentences to Wikipedia plot sentences. Their approach combines a tf-idf-based sentence similarity measure with a global alignment algorithm to achieve matches between the sentences of two documents. In particular, the authors employ a variant of the Needleman and Wunsch (1970) alignment algorithm, that dynamically finds the optimal alignments between sentences $1 \ldots i$ and $1 \ldots j$ of documents d_1 and d_2 respectively according to:

$$s(i,j) = \max \begin{cases} s(i-1,j-1) + p(match(i,j)) \\ s(i-1,j) + p(match(i,j)) \\ s(i,j-1) + p(match(i,j)) \end{cases}$$

This global alignment makes use of probability $p(match(i,j))$ of two sentences i and j matching each other. Similar to Nelken and Shieber (2006), we estimate this probability using a logistic regression model. The model is trained on a small number of manually-aligned Wikipedia-script instances, with features for word stem similarity as well as lemma overlap. When evaluated on four movies[6] whose content was manually aligned to Wikipedia plots, the aligner achieved a precision of .53 at a recall rate of .82

[6] "Cars 2", "Shrek", "Swordfish", and "The Silence of the Lambs".

1.	Why does Trevor leave New York and where does he move to?
2.	What is KOS, who is their leader, and why is he at school?
3.	What happened to Cesar's finger, how did he eventually die?
4.	Who killed Benny and how does Ellen find out?
5.	Who is Rita and what becomes of her?

Table 3.1: Examples of questions asked in human evaluation, for the movie "One Eight Seven".

at aligning script sentences to Wikipedia plots. To generate the gold standard, we assume that all scenes with at least one matching sentence in the Wikipedia plot is part of the full gold chain. In order to creating gold chains at different compression rates, we rank the scenes according to the number of alignments they contain. We then generate the compressed chain starting with the best-ranked scene and successively adding lower ranked ones until we reach the desired compression rate.

3.5.2 Evaluation

System Comparison In our experiments we compared our scene extraction model (SceneSum) against four baselines. The first baseline was based on the *minimum* character overlap (MinOv) of characters in consecutive scenes and corresponds closely to the diversity term in our objective. The second baseline was based on the *maximum* overlap (MaxOv) of characters and approximates the importance term in our objective. The third baseline (*n*th-scene) uniformly selects every *n*th scene based on the threshold. The fourth and final baseline selects scenes at random (averaged over 1,000 runs). Parameters for our models were tuned on the training set, weights for the terms in the objective were optimised to the following values: $\lambda_P = 1.0$, $\lambda_D = 0.3$, and $\lambda_I = 0.1$. We set the restart probability of our random walker to $\varepsilon = 0.5$, and the sigmoid scaling factor in our diversity term to $k = -1.2$.

We assessed the output of our model (and comparison systems) automatically against the gold chains described above. We performed experiments with compression rates in the range of 10% to 50% and measured performance in terms of F1. In addition, we also evaluated the quality of the extracted scenes as perceived by humans, which is necessary, given the approximate nature of our gold standard. We adopted a question-answering (Q&A) evaluation paradigm which has been used previously to evaluate summaries and document compression (Morris et al., 1992; Mani et al., 2002;

Clarke and Lapata, 2010). Under the assumption that the summary is to function as a replacement for the full script, we can measure the extent to which it can be used to find answers to questions which have been derived from the entire script and are representative of its core content. The more questions a hypothetical system can answer, the better it is at summarising the script as a whole.

Two annotators were independently instructed to read scripts (from our test set) and create Q&A pairs. The annotators generated questions relating to the plot of the movie and the development of its characters, requiring an unambiguous answer. They compared and revised their Q&A pairs until a common agreed-upon set of five questions per movie was reached (see Table 3.1 for an example). In addition, for every movie we asked subjects to name the main characters, and summarise its plot (in no more than four sentences). The full set of questions and corresponding expected answers are provided in Appendix B. Using Amazon Mechanical Turk (AMT)[7], we elicited answers for eight scripts (four comedies and four thrillers) under four summarization conditions: using our model, the two baselines based on minimum and maximum character overlap, and the random system. All models were assessed at the same compression rate of 20% which seems realistic in an actual application environment, e.g., computer aided summarization. The scripts were preselected in an earlier AMT study where participants were asked to declare whether they had seen the movies in our test set (65 in total). We chose the scripts which had received the least viewings so as to avoid eliciting answers based on familiarity with the movie. A total of 29 participants, all self-reported native English speakers, completed the Q&A task. The answers provided by the subjects were scored against an answer key. A correct answer was marked with a score of one, and zero otherwise. In cases where more answers were required per question, partial scores were awarded to each correct answer (e.g., 0.5). The score for a summary is the average of its question scores.

3.5.3 Results

Table 3.2 shows the performance of SceneSum, our scene extraction model, and the four comparison systems (MaxOv, MinOv, *n*th-scene, and Random) at five compression rates. As can be seen, MaxOv performs best in terms of F1, followed by SceneSum. We believe this is an artefact due to the way the gold standard was created. Scenes with large numbers of main characters are more likely to figure in Wikipedia

[7] https://www.mturk.com/

	10%	20%	30%	40%	50%
MaxOv	0.40	0.50	0.58	0.64	0.71
MinOv	0.13	0.27	0.40	0.53	0.66
*n*th-scene	0.11	0.21	0.3	0.4	0.5
SceneSum	0.23	0.37	0.50	0.60	0.68
Random	0.10	0.20	0.30	0.40	0.50

Table 3.2: Model performance on automatically generated gold standard (test set) at different compression rates.

	Beginning	Middle	End
MaxOv	33.95	34.89	31.16
MinOv	34.30	33.91	31.80
*n*th-scene	35.21	33.26	31.52
SceneSum	35.30	33.54	31.16
Random	34.30	33.91	31.80

Table 3.3: Average percentage of scenes taken from the beginning, middle and ends of movies, on automatic gold standard test set.

plot summaries and will thus be more frequently aligned. A chain based on maximum character overlap will focus on such scenes and will agree with the gold standard better compared to chains which take additional script properties into account. The *n*th-scene system shows a nearly perfect random result. Additionally, we check for a bias in the systems with respect to scene position. Tables 3.3 and 3.4 show the average percentage of scenes selected from the beginning, middle and end of movies, and the percentage of scenes selected from the first/last 10% of the movies' total scenes, respectively.

Table 3.3 indicates a slight bias of all systems towards selecting scenes earlier in the movie, but no one system seems to over-emphasise either part of the scripts. In contrast, Table 3.4 suggests a more expressed bias of *SceneSum* to select slightly more scenes from the first 10%, and slightly less from the last 10% of the scripts. It is unclear at this point in time whether this reflects an objective distribution of fewer key scenes at the last 10% of scripts in our test set, but note that the overall selectional preference in the last 33% does not show that divergence (Table 3.3). This means that *SceneSum* selects more scenes from the beginning of this portion. We leave an empirical evaluation of this behaviour to future work.

	First 10%	Last 10%
MaxOv	18.31	19.54
MinOv	17.69	17.23
*n*th-scene	20.00	20.00
SceneSum	22.00	16.92
Random	20.92	21.38

Table 3.4: Average percentage of scenes taken from the first/last 10% of a movie's scenes, on gold standard test set.

The results of our human evaluation study are summarised in Table 3.5. The *n*th-scene baseline was left out from the human experiments. We observe that SceneSum summaries are overall more informative compared to those created by the baselines. In two instances ("A Nightmare on Elm Street 3" and "Mumford"), the overlap models score better, however, in this case the movies largely consist of scenes with the same characters in them, with relatively little variation ("A Nightmare on Elm Street 3"), or the camera follows the main lead in his interactions with other characters ("Mumford"). Since our model is not so character-centric, it might be thrown off by non-character-based terms in its objective, leading to the selection of unfavourable scenes. Table 3.5 also presents a break down of the different types of questions answered by our participants. Again, we see that in most cases a larger percentage is answered correctly when reading SceneSum summaries.

Overall, we observe that SceneSum extracts chains which encapsulate important movie content across the board. We should point out that although our movies are broadly classified as comedies and thrillers, they have very different structure and content. For example, "Little Athens" has a very loose plotline, "Living in Oblivion" has multiple dream sequences, whereas "While She was Out" contains only a few characters and a series of important scenes towards the end. Despite this variety, SceneSum performs consistently better in our task-based evaluation.

3.6 Discussion and Conclusion

In this chapter, we presented *SceneSum*, a graph-based extractive summarisation system for movie scripts. We showed how to organise the information contained within a screenplay into meaningful graph-based and linguistic features, which can be exploited

Movies	MaxOv	MinOv	SceneSum	Random
Nightmare 3	69.18	**74.49**	60.24	56.33
Little Athens	34.92	31.75	**36.90**	33.33
Living in Oblivion	40.95	35.00	**60.00**	30.24
Mumford	**72.86**	60.00	30.00	54.29
One Eight Seven	47.30	38.89	**67.86**	30.16
Anniversary Party	45.39	56.35	**62.46**	37.62
We Own the Night	28.57	32.14	**52.86**	28.57
While She Was Out	72.86	75.71	**85.00**	45.71
All Questions	51.51	50.54	**56.91**	39.53
Five Questions	51.00	53.13	**57.38**	36.88
Plot Question	60.00	56.88	**73.75**	55.00
Characters Question	**45.54**	37.34	37.75	31.29

Table 3.5: Percentage of questions answered correctly in human evaluation.

in a dynamic programming algorithm for scene extraction. The overall problem is split into a variety of objectives, which are jointly optimised in a global objective function, taking into account the scene importance, diversity and progression of a generated summary. Automatic as well as Human evaluation studies suggest that *SceneSum* generates informative summaries of screenplays, which outperform several baselines for the task.

The nature of the objective function also makes *SceneSum* very adaptable to different scenarios. Changing weights in the global objective function, we can generate summaries that pertain to specific aspects of screenplays, for example focusing on character progressions. It is also easily extensible, by adapting and adding aspects to either the global objective function, or its individual terms.

A further point regarding the individual objective weights is the influence of *genre* and other aspects of the movie on the optimal parameter setting. In our experiments, we optimised parameters broadly to cover all movie scripts in our dataset, disregarding differences between certain films. It is conceivable, however, that aspects such as a movie's genre can have an influence on which extraction parameters – *progression*, *diversity*, or *importance* – should be regarded as more/less influential than others. For example, in a movie like *"Marvel's Avengers - Civil War"*, a fast-paced superhero ensemble movie featuring a great variety of characters, we might want to put higher

emphasis on the *diversity* term, to make sure the extracted scenes capture aspects of all the characters' story lines. On the other hand, in a slower movie focusing on fewer characters such as the romantic comedy-drama "The Terminal" starring Tom Hanks and Catherine Zeta-Jones, the optimal system likely can almost disregard the *diversity* term as the film is about the two protagonists, and not many other characters are present. Instead, the *importance* and *progression* terms might contribute more to a "good" summary.

One potential drawback of the approach presented here is the computational complexity of the linear problem at hand. Given that we aim to summarise entire screenplays, often consisting of 100 and more scenes, the number of variables that have to be computed between each pair of scenes rises exponentially. This leads to linear problems with tens of thousands of variables and constraints. We therefore limited experiments to systems that have exhaustively optimised *global weights* for progression, diversity and importance terms. Ideally, we would optimise each micro-term individually to further enhance system performance. One potential way of overcoming the challenge this local optimisation poses would be to constrain the model constraints. For example, it is unlikely if not impossible for an actual extracted summary to jump extreme distances, such as from the very beginning of the movie to the very end, excluding all middle parts. We could heuristically exclude variables and constraints corresponding to such unlikely extraction events, which could potentially greatly reduce their number. On the other hand, imposing such constraints means a great deal more manual engineering. Furthermore, it potentially limits system performance on "strange" movies, for example if they have an unexpected narrative structure.

One specific point of the extraction model we would like to improve on is the way sentiment scores are derived, for character-character sentiments, and the overall sentiment of scenes and movies. The current approach of utilising the AFINN lexicon to score sentences is very approximate, but still showed the best performance at the time of system development. In recent years, some progress has been made in estimating sentiments for sentences, for example using deep neural networks (Socher et al., 2013). Such advances could prove beneficial for our scene extraction model, and should be integrable as a replacement sentiment estimator with relatively little effort.

Furthermore, the current system evaluation was limited to a rather small set of genre-specific movies. In the future, we plan to explore model performance in a wider range of movie genres, and a greater number of movies in general. Additionally, *SceneSum* promises to be applicable to other NLP tasks which are structurally similar,

such as theatre play or book summarisation. Another route we would also like to pursue is the automatic determination of the compression rate, which should presumably vary according to the movie's length and content.

Finally, one major task we are concerned with is the adaptation of *SceneSum* to other domains, and incorporation of multi-modal constraints. We would, for example, expect the *musical score* or general sound effects of a movie to be quite indicative of certain events of interest. Similarly, visual features obtained from a movie's video are expected to provide valuable information for the scene extraction process.

In the next chapter, we take a first step toward such an extended system, adapting *SceneSum* to the film domain, and introducing multi-modal setups that exploit both text and visual information.

Chapter 4

Movie Summarisation via Multi-modal Scene Extraction

Movie Summarisation is the task of generating a shorter version of a full length feature film, while maintaining the important information and story arcs of the movie. As opposed to trailer generation, a good summary of a movie has to contain all its important parts, covering the film from beginning to end, and leave the viewer knowing about everything that has happened in the movie. Depending on the allowed length of the summary, the resulting shortened version can be used to skim the film in order to find out if it is interesting, to present a "what happened so far" for a previous movie in a series of films, or present potential viewers with a short gist of what is about to come. We propose to model the problem of movie summarisation from a scene extraction perspective, which uses a variety of multi-modal features derived from the actual movie as well as the corresponding script, extending the previously introduced character-graph based script scene extraction model. We show how we obtain features from films that are readily integrated into the graph-based scene extraction model, either replacing the original script-based objectives, or complementing them in a multi-modal setting. While preliminary experiments do not seem to favour a system that makes use of information provided by all available modalities, user studies using a question answering task show that the summaries produced by the multi-modal system are informative, and enable users to acquire more knowledge about important plot points than from summaries generated by baseline systems.

4.1 Introduction

More than 700 movies are released in theatres each year, according to Boxoffice Mojo[1], a website tracking the theatrical release and box office performance of feature films. An even greater number of films sees publication for TV, home cinema, or online platforms such as Netflix[2] or Hulu[3]. Given this large amount of releases, many tasks arising during a film's life-cycle stand to benefit from automated or semi-automated (computer aided) approaches to process the data. For example, in order to make a particular film attractive to audiences, studios typically release teasers and trailers, very short clips of not more than 180 seconds designed to advertise the movie. In other scenarios such as multi-part movie releases – which often see films of the same series released a number of years apart – studios, cinemas or other providers might want to offer full-fledged summaries of previous instalments, to remind viewers of "what happened so far" and lead into the new release. A similar argument can be made for summaries covering the entire series. Similarly, on-demand film platforms might make use of film summarisation techniques to either provide short summaries users can watch before committing to a full movie, or they might even offer their users a more dynamic viewing experience, for example by letting them skim through films with focus on certain aspects (e.g., "show me all action scenes").

The previous chapter explored how the task of summarising a movie's *script* can be addressed as a scene extraction problem, utilising a social graph induced over the screenplay. A natural follow-up question arising from the previous findings is whether we can summarise the *actual movie*, employing a similar approach. In this chapter, we address this task of feature film summarisation, by developing a multi-modal variant of our original *SceneSum* model. Using dynamic time warping (Berndt and Clifford, 1994), we show how to obtain scene-segmentations for movies through the alignment of subtitle data to the script text which provides information about scene boundaries. For a selection of scripts contained in *ScriptBase*-J, we process the actual movie to obtain visual features. We show how the previous notions of *progression, diversity*, and *importance* can be applied to the visual domain by obtaining features based on motion pictures, and how they can be combined with script-based features to obtain a multi-modal scene extraction model. Finally, we generate visual summaries of movies and conduct human evaluation studies similar to the script summarisation case. Results

[1] http://www.boxofficemojo.com/
[2] www.netflix.com
[3] www.hulu.com

show that the multi-modal visual scene extraction method outperforms several baseline systems.

4.2 Related Work

Video summarisation is a long-standing field of research in Computer Vision and other communities (for overviews, see Reed 2004; Money and Agius 2008). A variety of techniques have been employed for the task, summarising different kinds of video. Lu and Grauman (2013) summarise egocentric videos by detecting and tracking visual objects. Many approaches use features directly derived from the source video, audio or textual information such as subtitles, or from multi-modal combinations thereof. In Ma et al. 2002, 2005, the authors model users attention levels based on audio-visual features, generating summaries that are predicted to reflect high levels of interest for viewers. More recently, in a similar approach, Evangelopoulos et al. (2013) summarise movies by detecting events and fusing shots based on saliency features derived from video, audio, and text. In the film production setting, Wang and Ngo (2007) use object detection, camera motion and speech clips to filter *rushes*, long raw cuts of movies containing repetitions and junk shots. Approaches which abstract away from the underlying source video include Sang and Xu (2010b), who identify characters in video, and take character similarity graphs into account when generating summaries for movies. Lin et al. (2013) follow a similar approach, summarising movies with the help of character-character networks. In a very different multi-modal setup, Ren et al. (2010) relate Wikipedia entries to films by inducing and matching topics between video and text using Latent Dirichlet Allocation. They then generate summaries over the induced topics, by modelling user attention.

Our own model represents an extension of the script-based scene extraction model described in Chapter 3, Gorinski and Lapata (2015), itself building on the techniques presented in Lu and Grauman (2013). It extends the original text based model by features derived from the source video. In contrast to most previous work, the model is built around the central notion of *scenes*, as opposed to shots, and aims at summarising entire feature films.

4.3 Multi-Modal Movie Summarisation

When summarising a movie, there are various sources that can be considered for informing the summarisation process. As opposed to the script-base case presented earlier, when dealing with the actual film, a great deal more information is available, such as subtitles, audio cues, and of course the video itself. All these modalities can offer potential benefits for the scene extraction process: Scenes that may seem uninteresting in a textual description, e.g., if the whole scene only consist of a short description like

```
``We see the hero's parents in a dark alley, waiting for a taxi
when suddenly -- a shot -- the father sinks to the ground.''
```

may seem short and unimportant in the script setting, even though it might provide an important insight into the hero's backstory. However, if this scene is filmed in a visually interesting manner, and potentially riddled with audio cues, a system that takes such features into account may be able to pick up on the impact of this short segment, and include it in a summary. Conversely, scenes that may seem visually uninteresting can contain dialogue that is vital to the film's story. However, this could only be picked up by a system that has access to the characters' dialogue, by considering either the script, subtitles, or maybe audible speech.

For a task like movie summarisation, *multi-modal* systems are therefore an immensely interesting area of research, allowing to access and incorporate mutually exclusive information from a variety of sources. While the ideal system would surely include all possible modalities, in this chapter we concentrate on adding the *video* of a feature film as an additional modality to the script-based scene extraction model of Chapter 3.

4.4 Visual Scene Extraction Model

Analogously to the graph-based scene extraction model presented in Chapter 3, we define the task of movie summarisation as a scene extraction problem. Given a movie M consisting of an ordered set $S = (s_1, s_2, \ldots, s_n)$ of scenes, we want to find a summary represented by the chain $S' = (s_i, \ldots, s_k); 1 \leq i, k \leq n$ of consecutive scenes subject to a compression rate m, which simultaneously optimises for a smooth scene progression,

diversity, and overall scene importance:

$$S^* = \underset{S' \subset S}{arg\,max}\, Q(S') \tag{4.1}$$

$$Q(S') = \lambda_P P(S') + \lambda_D D(S') + \lambda_I I(S') \tag{4.2}$$

We previously defined the progression, diversity and importance terms P, D and I based on features which we obtained from the movie screenplay and its induced character-character graph, such as identifying the main and support characters, modelling their interactions, or calculating sentiment scores for scenes. As this information is not readily available in the video domain, we here re-define these objective terms to incorporate information from a movie's source video. A visual model is important, for example in cases where the script is not available, or where NLP tools for script processing are lacking, e.g., for non-English languages.

Scene-to-scene Progression For the progression term of the global objective function, we keep the intuition of measuring the characters' influence of moving from scene to scene through the selected scene chain. However, as character information is not directly available from the video source, we re-define the original bipartite graph as ranging over scenes on the one hand, and *face clusters* on the other. By identifying and clustering faces over the whole length of the movie, we can approximate the characters of the film without relying on external sources like the script or possibly audio to identify them. We can then define the progression score of a scene chain like before, as the sum influence of clusters on the progression between two scenes through a bipartite graph B:

$$B = (V, E) \,:\, V = C \cup S$$

$$E = \{(s, c, w_{s,c}) | s \in S,\, c \in C,\, w_{s,c} \in [0,1]\} \,\cup\, \{(c, s, w_{c,s}) | c \in C,\, s \in S,\, w_{c,s} \in [0,1]\}$$

where S and C denote scenes and *face clusters*, respectively.

We need to re-define the edge weights in B in terms of weights between scenes and clusters. Unlike before, we cannot rely on main character identification via a trained classifier, and approximate the weights $w_{c,s}$ from clusters to scenes as the "pseudo" probability P^* of the cluster belonging to a main character of a movie.

$$w_{c,s} = P^*(c \in main(M)), \forall (c, s, w_{c,s}) \in E \tag{4.3}$$

We keep the original definitions from Chapter 3 for edge weights $w_{s,c}$ from scenes to clusters, as being a measure of the relative involvement of a character in all interactions occurring in a scene (see Section 4.6 for our definition of interactions in the

visual modality):

$$w_{s,c} = \frac{\sum_{c' \in C_s} inter(c,c')}{\sum_{c_1,c_2 \in C_s \times C_s} inter(c_1,c_2)}, \forall (s,c,w_{s,c}) \in E \tag{4.4}$$

Similarly, we keep the notion of the influence of a cluster c on the progression from scene s_i to scene s_{i+1} as being reflected by the change of the probability that a random walker starting in s_i of the bipartite graph ends up in s_{i+1}, when removing c from the graph:

$$INF(s_i, s_{i+1}|c) = r_{s_i}[s_{i+1}] - r^c_{s_i}[s_{i+1}] \tag{4.5}$$

As before, the overall scene-to-scene progression of a chain S' is the sum of all its influence scores

$$P(S') = \sum_{i=0}^{|S'|-1} \sum_{c \in C_i} INF(s_i, s_{i+1}|c) \tag{4.6}$$

We refer to the next sections to show how we obtain face clusters to model characters and their interactions for a motion picture.

Diversity For the diversity term $D(S')$, we want to measure the dissimilarity of two movie scenes connected in the chain. Analogous to the scrip-based model, we define the overall diversity as:

$$D(S') = \sum_{i=1}^{|S'|-1} d(s_i, s_{i+1}) \tag{4.7}$$

Due to the nature of the source material, we have to reconsider the definition of $d(s_i, s_{i+1})$. To this end, we take four factors into account: (a) The face clusters shared between the scenes, (b) the scenes' difference in length, (c) their difference in number of shots included, and (d) the scene's difference in the number of objects they contain:

$$d(s_i, si+1) = \frac{d_{clust}(s_i, s_{i+1}) + d_{len}(s_i, s_{i+1}) + d_{shots}(s_i, s_{i+1}) + d_{obj}(s_i, s_{i+1})}{4} \tag{4.8}$$

where $d_{clust}(s_i, s_{i+1})$, $d_{len}(s_i, s_{i+1})$, $d_{shots}(s_i, s_{i+1})$, and $d_{obj}(s_i, s_{i+1})$ are the various terms considered with respect to scene difference.

The difference of clusters between two scenes is easily defined as:

$$d_{clust}(s_i, s_{i+1}) = 1 - \frac{|C_{s_i} \cap C_{s_{i+1}}|}{|C_{s_i} \cup C_{s_{i+1}}|} \tag{4.9}$$

As with character overlap in the original model, d_{clust} will be 0 if the scenes share all clusters, and 1 if they contain entirely disjoint sets.

To measure the difference in terms of length, we define d_{len} as:

$$d_{len}(s_i, si+1) = 1 - \frac{min(len(s_i), len(s_{i+1}))}{max(len(s_i), len(s_{i+1}))} \tag{4.10}$$

Here, $len(s')$ denotes the runtime of a scene. d_{len} becomes 0 if the two scenes are equal in length, and tends toward 1 the bigger their difference.

For the third term, we measure the numbers of shots contained in either scene analogous to d_{len}:

$$d_{shots}(s_i, si+1) = 1 - \frac{min(|shots(s_i)|, |shots(s_{i+1})|)}{max(|shots(s_i)|, |shots(s_{i+1})|)} \tag{4.11}$$

where $shots(s')$ is the set of all shots contained in s'. Intuitively, it may seem redundant to measure both a scene's length in time as well as the number of shots it is comprised of however, the two do not necessarily correlate. For example, a relatively short action scene may contain a large number of fast, staccato-like shots, while a long but sombre scene in the same film may contain just one or two.

Our final diversity term measures the difference of two scenes with respect to their "busyness". We identify salient objects in each scene, and take their average count as an indication of how busy a scene seems to be. We can then define the relative difference between two scenes as:

$$d_{obj}(s_i, s_{i+1}) = 1 - \frac{min(avob(s_i), avob(s_{i+1}))}{max(avob(s_i), avob(s_{i+1}))} \tag{4.12}$$

$$avob(s) = \frac{\sum_{f=0}^{F} obj(s^f)}{\sum_{i=0}^{F} 1} \tag{4.13}$$

where s^f denotes the *f-th* frame of a scene, and F is the total number of frames comprising the scene, and $obj(s^f)$ gives the number of objects identified in frame f.

Importance The third term $I(S')$ in the global objective captures the overall importance of a single scene. In the film domain, we define that importance as:

$$I(S') = \sum_{i=1}^{|S'|} imp(s_i) \tag{4.14}$$

Similar to scene diversity, we take into account several importance measures, reflecting a scene's importance with respect to (a) face clusters, (b) the scene's length, (c) the number of contained shots, and (d) salient objects found in the scene:

$$imp(s_i) = \frac{imp_{clust}(s_i) + imp_{len}(s_i) + imp_{shots}(s_i) + imp_{obj}(s_i)}{4} \tag{4.15}$$

In terms of face clusters present in scenes, we assume a scene is more important the more clusters it contains:

$$imp_{clust}(s_i) = \frac{|C_{s_i}|}{\max\limits_{s_j \in S} |C_{s_j}|} \tag{4.16}$$

The second term accounts for a scene's runtime, as its fraction of the total runtime of the movie:

$$imp_{len}(s_i) = \frac{len(s_i)}{\sum_{j=0}^{|S|} len(s_j)} \tag{4.17}$$

Analogous to diversity, in addition to runtime, we also capture scene importance with respect to the number of shots contained in it as:

$$imp_{shots}(s_i) = \frac{|shots(s_i)|}{\max\limits_{s_j \in S} |shots(s_j)|} \tag{4.18}$$

Finally, we measure the busyness of a scene as the average number of detected salient objects, relative to the maximum number of objects encountered in any frame:

$$imp_{obj}(s_i) = \frac{avob(s_i)}{\max\limits_{f} obj(s^f)} \tag{4.19}$$

The optimal chain of scenes that maximises the progression, diversity and importance of extracted scenes is selected analogously to the script-based case, using linear programming to maximise Equation (4.2). We discuss how we obtained the features required by our film-based chain extraction framework in Section 4.6.

4.5 Multi-modality

All terms defined in this section so far are specific to the video domain. However, due to the nature of our original scene extraction model, these video-based terms can be readily integrated into a combined *multi-modal* scene extraction model for movies, by combining the individual sub-terms presented here and in the previous chapter. To this end, in our multi-modal setup, we combine the *progression*, *diversity* and *importance* terms for the script and film domains into larger terms of the global objective function:

$$S^* = \underset{S' \subset S}{arg\,max}\, Q_{multi}(S') \tag{4.20}$$

$$Q_{multi}(S') = \lambda_{P_{multi}} P_{multi}(S') + \lambda_{D_{multi}} D_{multi}(S') + \lambda_{I_{multi}} I_{multi}(S') \quad (4.21)$$

$$P_{multi}(S') = P_{script}(S') + P_{video}(S') \quad (4.22)$$

$$D_{multi}(S') = D_{script}(S') + D_{video}(S') \quad (4.23)$$

$$I_{multi}(S') = I_{script}(S') + I_{video}(S') \quad (4.24)$$

This model represents the straight-forward linear combination of the script-based system with the video-based objectives. The new combined objective function Q_{multi} consists of the combined *progression*, *diversity* and *importance* terms of both systems. The terms are weighted by globally through weights λ.

An advantage of this approach is that like in the original setting, we need only be concerned with estimating a small set of parameters for the extraction process. However, it might be desirable to weigh each sub-term individually. While we chose to forego this possibility (see also discussion in Section 4.9), this change is trivial to implement in the system, by introducing individual parameters λ for the sub-terms of Equations (4.22)–(4.24), or even for the individual terms that are used to calculate diversity and importance.

4.6 Implementation

4.6.1 Scene Boundary Identification

For the model presented in this thesis, the *scene* is the central unit considered in the extraction process. While scenes can be readily approximated in the script-based case by observing scene headings (as shown in Figure 3.1), the task of scene identification is much harder in the visual domain. Mediums like DVDs typically contain *chapter* information however, in most cases these chapters do not correspond to the scenes of the movie, but rather comprise multiple scenes that form a larger unit. On the other hand, movies make it comparatively easy to identify *shots*, for example using shot segmentation via colour histograms (Kasturi et al., 1996) or pixel-to-pixel differences. For an overview of various segmentation models, see Dailianas et al. (1995). Regardless of the segmentation method, *shots* typically constitute a significantly smaller unit than *scenes*, which are usually made up of several shots. Both, chapters and shots are therefore not directly suited for our scene extraction task.

Sankar et al. (2009) propose a scene detection model working on visual and audio-cues from the source video alone. While being helpful in situations where additional

information is not available, their method is also prone to misidentify scene boundaries. Another approach is to obtain scene boundaries via script alignments with movie subtitles or closed captions (Cour et al. 2008; Park et al. 2010). We choose this option, as it promises higher precision scene boundary identification. Movie scripts are available to us through *ScriptBase*-J, and subtitles for use in script-subtitle alignment are nowadays readily obtainable either from DVDs or online, via sites like Open Subtitles[4].

Shot Detection Even though shots are a smaller unit than scenes, having information about shot boundaries can still be advantageous. In addition to the diversity and importance terms in our model which rely in part on the number of shots in a scene, we might also want to exploit shot segmentation information during the scene segmentation for the movie. For example, scene boundaries should in all cases coincide with shot boundaries, with the last frame of a particular shot ending a scene, and the first frame of the next shot starting the new scene. We therefore need to be able to identify shot boundaries for the movies in our dataset.

A variety of shot boundary detection methods are available, as outlined in Dailianas et al. (1995). They range from pixel-wise comparisons of two images, to segmentation based on edge-detections however, histogram-based methods are widely considered the simplest and most effective. A histogram $H(f,k)$ of an image f separates the image into k bins of pixel intensities, e.g., 256 possible values for a grey-scale image. Examples of some images and associated histograms are shown in Figure 4.1.

Given the histograms of two images f and f', we can compute the *simple histogram difference* between them as:

$$d(f,f') = \sum_{j=0}^{k} |H(f,j) - H(f',j)| \tag{4.25}$$

In other words, we compute the sum of absolute differences between each bin of the images. Two images can be considered as belonging to the same shot if their difference is below a certain threshold t, or as signifying a shot boundary if it exceeds that threshold. Even though more involved histogram-based methods are available (Dailianas et al., 1995), manual inspection of shots detected for our test movies did not seem to favour other methods over a simple histogram difference.

Script-Subtitle Alignments via Dynamic Time Warping In order to obtain *scene boundaries* for movies, we follow an approach similar to Cour et al. (2008) and Park

[4] www.opensubtitles.org

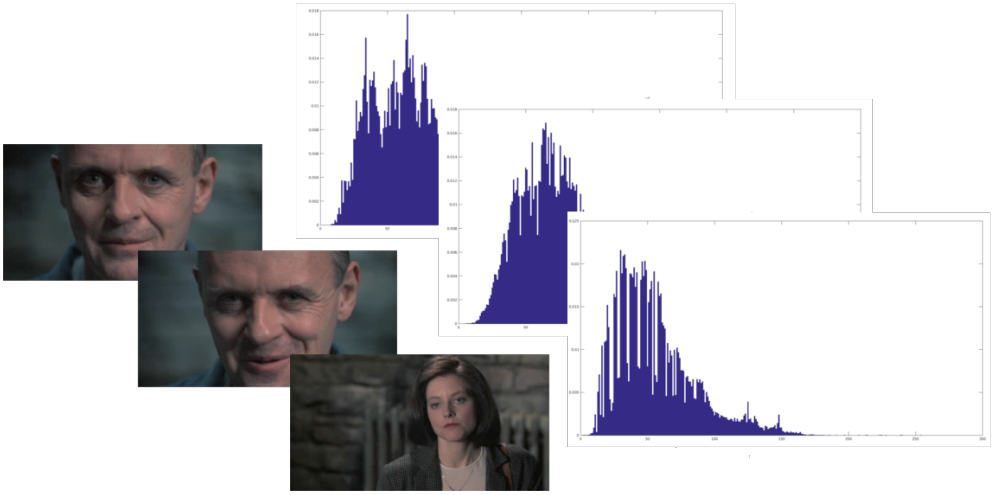


Figure 4.1: Captured frames and corresponding colour histograms, from the movie “The Silence of the Lambs”.

et al. (2010), and align movie scripts to the subtitles of the corresponding feature film. An example of the structure of subtitle files is shown in the top part of Figure 4.2. As can be seen, the subtitles provide detailed timestamps, indicating at which times of the movie the associated phrases are uttered. Comparing this subtitle information to the corresponding parts of the movie’s script, shown in the bottom part of Figure 4.2, one can see how the subtitles match the script. By aligning the two texts, we can therefore respectively obtain information about which scenes the movie’s subtitles belong to, and what time frame of the movie the screenplay’s scenes cover.

One potential problem is the imperfect alignment between subtitles and script text. As shown in Figure 4.2, the script may contain additional dialogue that did not make it into the final movie. Similarly, some movies exhibit ad-libbed dialogue that was not present in the shooting script during production. We thus may have to skip portions of either text during alignment, in order to overcome these dissimilarities. Dynamic time warping (Berndt and Clifford, 1994), DTW, seems a natural fit for this sort of alignment problem. Formally, if $A = a_1,\ a_2,\ \ldots,\ a_n$ and $B = b_1,\ b_2,\ \ldots,\ b_m$ are time series of respective lengths n and m, and D is an $n \times m$ matrix with distance values between elements of A and B such that $D_{i,j} = d(a_i, b_j)$, the dynamic time warping

```
00:13:46,790 --> 00:13:49,141
You use Evyan skin cream.

00:13:51,710 --> 00:13:54,540
And sometimes you wear L'Air du Temps.

00:13:55,830 --> 00:13:58,180
But not today.

00:13:59,470 --> 00:14:02,031
Did you do all these drawings, Doctor?
```

```
:SC: INT. DR. LECTER'S CORRIDOR - DAY

[...]

                              DR. LECTER
                    I see.  I myself cannot.  You use Evyan
                      skin cream, and sometimes you wear
                      L'Air du Temps, but not today.  You
                   brought your best bag, though, didn't you?

                              CLARICE
                               (beat)
                               Yes.

                              DR. LECTER
                         It's much better than your shoes.

                              CLARICE
                        Maybe they'll catch up.

                              DR. LECTER
                        I have no doubt of it.

                              CLARICE
                        (shifting uncomfortably)
                      Did you do those drawings, Doctor?
```

Figure 4.2: Example subtitles for a short segment of "Silence of the Lambs" (top), and corresponding portion of the screenplay text (bottom).

algorithm will find the optimal path W^* through D such that:

$$\begin{aligned} W^* &= \underset{W}{\operatorname{argmin}} \left(\frac{1}{K} \sqrt{\sum_{k=1}^{K} d_k} \right) \\ W &= w_1,\ w_2,\ \ldots,\ w_k \\ max(m,n) &\leq K \leq m+n1 \end{aligned} \tag{4.26}$$

where each K is the total length of the found path, $w_k = (i, j)$ indexes into elements in D. The resulting alignment is the one that minimises the overall distance between the aligned segments of the time series.

For the task at hand, we can naturally treat the (dialogue) sentences of the original screenplays and the subtitles of the movie as the time series A and B to be aligned. The distance function to generate the distance matrix D can be interpreted as a text-edit distance between sentences in the script and subtitles. In our setup, we make use of the Levenshtein distance (Levenshtein, 1966) between sentences [5].

4.6.2 Salient Object Detection

The importance and diversity terms of our scene selection model require a count of visual objects, to give an estimate of how busy a scene is (see Equations (4.12) and (4.19)). The task of identifying objects in video is known as *salient object detection* or *object segmentation*. A plethora of systems have been published for this task, for an overview of different models and specialised tasks, see Borji et al. (2015).

For the scene extraction model presented here, the key considerations that have to be taken into account are robustness, and the time it takes to segment the objects. Robustness is very important as we are dealing with unconstrained video, with a completely open class of objects being shown on screen. For the same reason, we are not concerned with *object identification*, i.e., the actual classification of a segmented object as belonging to one of several possible classes like *cat, dog,* or *car*. Instead, we are only interested in assigning "objectness" to parts of an image. Secondly, the runtime of the segmentation model is crucial, as we are processing entire feature films.

[5] The Levenshtein distance $lev_{a,b}(|a|,|b|)$ between two strings a and b is recursively defined as:

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & if min(i,j) = 0, \\ min(i,j) \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & otherwise \end{cases}$$

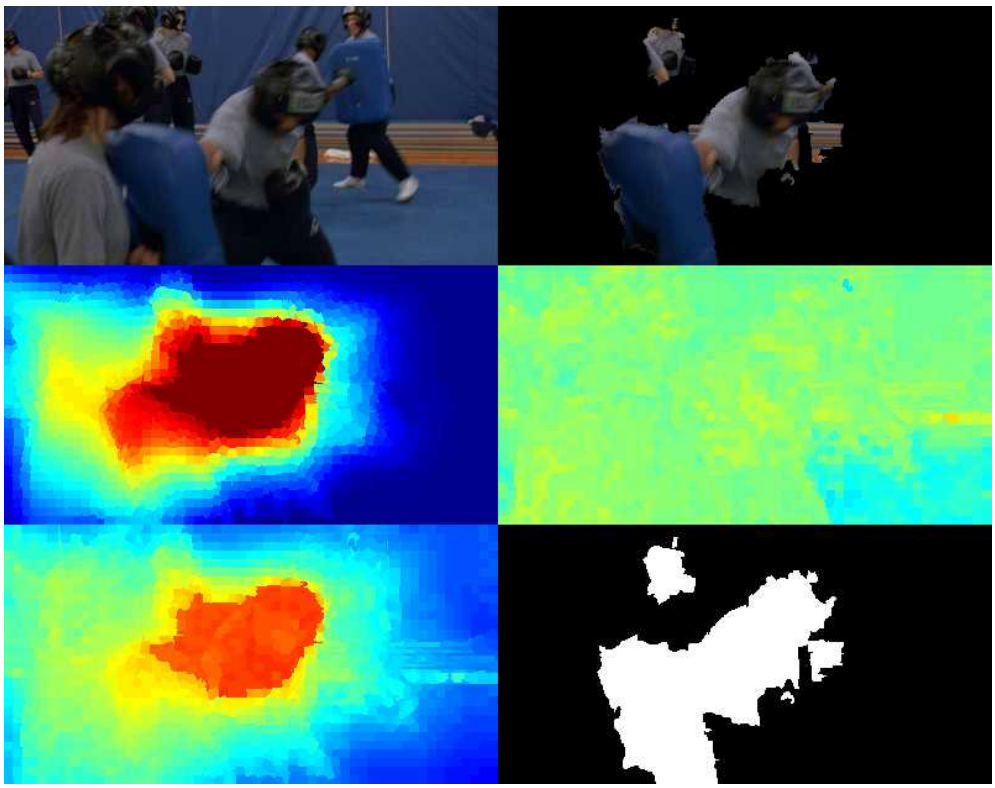


Figure 4.3: Sample object segmentation output of one frame from “The Silence of the Lambs”, with original image, objectness heat maps and segments. The segmentation identifies 4 visual objects in this frame.

These provide *hours* worth of video at a time, as opposed to mere minutes or even only seconds found in many segmentation datasets.

For these reasons, we opt for the segmentation model presented in Papazoglou and Ferrari (2013), which offers fast object segmentation in unconstrained video. Their model uses optical flow (Brox and Malik, 2010), the frame-to-frame displacement of pixels in the video, to detect motion boundaries (Sundberg et al., 2011), i.e., regions of the image that move at a different speed from the rest. The resulting boundaries are then refined via inside-outside maps between pixels and the polygons containing the bounded regions to separate objects. The model’s implementation has been made public[6], and is both fast and robust. We use their model to segment objects within each shot of a movie, and assign the number of segmented objects as a feature for each frame. An example of a visualised segmentation is shown in Figure 4.3. Note that anything can be a “visual object”, such as a punching matte or a person.

[6]available at `http://calvin.inf.ed.ac.uk/software/fast-video-segmentation/`

4.6.3 Face Networks

Analogous to *characters* in the script-based setting, our proposed film-based scene extraction relies on networks between recurring entities in the movie. In the script-based case, these entities were naturally taken to be the characters[7] that are encountered in the movie, and the network was constructed through their interactions in the script. In the film domain, information about the (named) characters is not available, as the frames of the video do not directly contain it. However, it is reasonable to assume that in any given movie, one actor plays exactly one character. It should therefore be possible to reconstruct character information by detecting faces in the video, and associate the faces of the same actor detected in different frames with one character.

In order to achieve this character detection, we employ a two-stage process of *face detection* and *face clustering* which enables us to generate an interaction network between the identified clusters, under the assumption that clusters represent characters.

Face Detection and Tracking *Face detection* is the task of identifying parts of an image that constitute a face. It has been an active area of research for many years, and there exist a large amount of standard models for the task such as the Viola-Jones algorithm (Viola and Jones, 2001, 2004). Many of these standard implementations pose limitations that are too severe for our domain. For example, there exist separate models specialising on portrait or profile face detection, or even detecting rotated faces. Of course all of these types, and potentially more, are present in a free-form feature film.

Mathias et al. (2014) recently published a fast, robust face detection model that is able to handle a large variety of facial positions, without sacrificing detection quality. These properties make it very well suited for application in the film domain. The face detection framework is based on the Deformable Parts Model (Felzenszwalb et al., 2010), which is recognised as being both robust and relatively fast. As we do not have training data specific to our task available, we instead rely on the freely available implementation and face models by the original authors[8].

In addition to detecting faces on a frame-by-frame basis, we also employ *face tracking* in order to identify across frames of the same shot. We employ an implementation of the Kanade-Lucas-Tomasi (KLT) algorithm (Lucas and Kanade, 1981; Tomasi and Kanade, 1991). KLT uses point features of faces detected in consecutive frames in

[7] We considered only characters with a talking role in the script.

[8] available at `http://www.markusmathias.de/face_detection/`

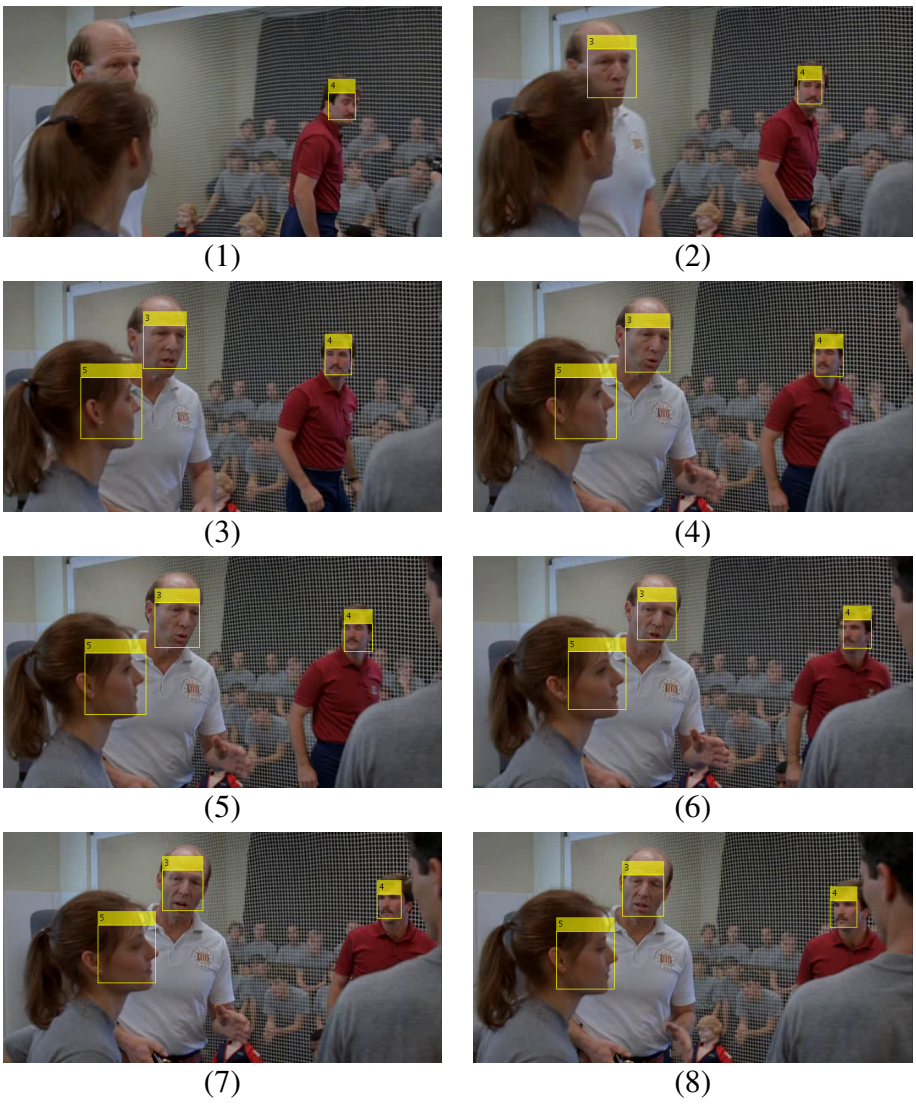


Figure 4.4: Example of face detection and tracking, on a short segment of "The Silence of the Lambs". Boxes are around detected faces. The number correspond to the track the detected faces are assigned to.

order to determine whether two detections belong to the same track, and allows for tracking multiple faces in the same shot. Figure 4.4 shows an example of multiple faces being detected and tracked through a short sequence of a shot.

Face Clustering The final pre-processing step for constructing face networks for movies is *automatic face clustering*. As we are unable to directly identify the characters present in a movie from the source video alone, we rely on the assumption that we can infer characters from automatically generated clusters. To this end, we employ a constrained clustering approach based on Hidden Markov Random Fields, presented

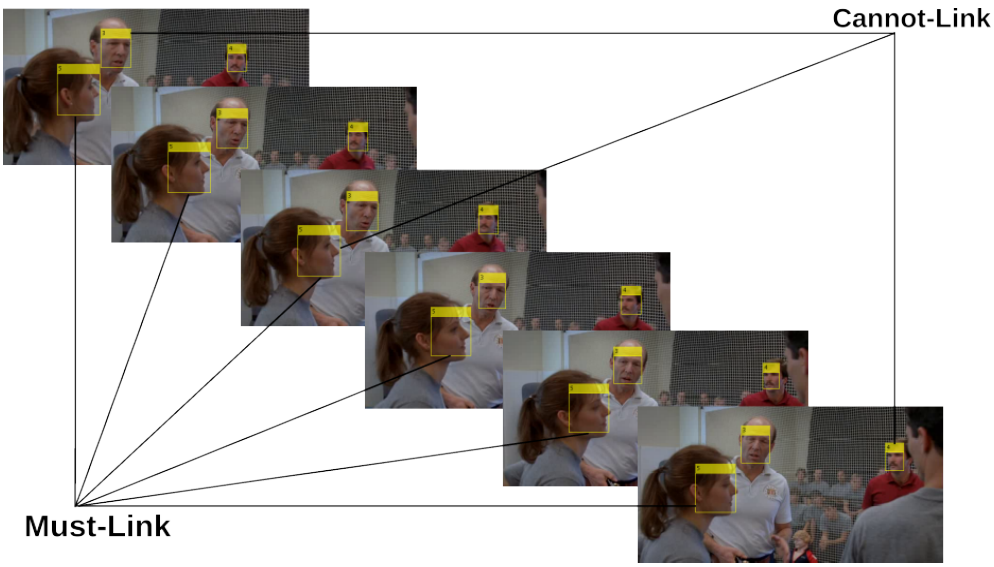


Figure 4.5: Illustration of must-link and cannot-link constraints for a small sample of frames from a shot taken from “The Silence of the Lambs”. Faces from the same track have to be clustered together, while faces that belong to different tracks have to be assigned to different clusters.

in Wu et al. (2013).

The central idea of the clustering algorithm is the introduction of two constraints that guide the clustering process, for faces that *must be linked* and faces that *cannot be linked*. Instead of simply clustering all faces that have been identified in the source video, the two constraints make use of the *face tracks* established in the previous processing step, and make sure that (i) faces belonging to the same track *must* be in the same cluster, and (ii) faces belonging different tracks *cannot* belong to the same cluster. Figure 4.5 illustrates the two constraints.

We extract features for the sampled faces using the deep feature network of Parkhi et al. (2015), implemented on top of MatConvNet (Vedaldi and Lenc, 2015). We then cluster the faces of the film using the constrained clustering approach.

An advantage of this clustering method is that the final clusters promise to be purer than ones that were generated without constraints, as it is guaranteed that faces from the same track are clustered together, and faces from different tracks are separated. However, it also introduces significant computational costs for the objective function that solves the cluster assignment problem. This cost is exacerbated in the movie case, as we are dealing with full-length feature films with hundred-thousands of frames, constituting thousands of shots, each potentially containing a large number of faces.

Using every single detected face during clustering is therefore intractable in our setup. Instead, we sample up to 5 faces randomly from each identified track and cluster those samples, greatly reducing computational demands.

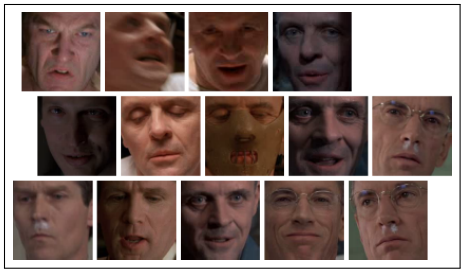
In addition to the sampling of faces from face tracks, the clustering algorithm of Wu et al. (2013) also needs to know a-priory how many clusters exist, for each movie. To this end, we could employ outside sources, such as Wikipedia or the movie script, to determine the number of characters that appear in the film. However, for the visual features in our system, we do not want to have to rely on extra-visual information. We therefore employ the following method to determine the number of output clusters: First, we set the number of clusters to be equal to the *maximum number* of faces detected in any frame of the movie, and apply the constrained clustering algorithm. In cases where the constraints cannot be satisfied, i.e., not all faces that *must* be linked can be assigned to the same cluster, and not all faces that *cannot* be linked can be assigned to different clusters, we incrementally increase the number of clusters and apply the algorithm again, until the constraints can be satisfied. We show examples for some clustering results in Figure 4.6.

Cluster Networks Using the clusters established in the previous steps, we finally are able to construct cluster networks and bipartite scene-cluster graphs parallel to the character networks and scene-character graphs of Chapter 3, needed for the scene extraction model. We generate cluster graphs by taking clusters as vertices, and create edges between them according to cluster interactions. We count as an interaction any time two clusters appear together in the same shot, and set the edge-weights between vertices in the cluster graph to reflect the total number of interactions they share throughout the movie. We then construct the bipartite scene-cluster graph as outlined in Section 4.4, setting edge-weights $w_{s,c}$ from scenes to clusters as by Equation (4.4).

In order to set weights $w_{c,s}$ from clusters to scenes, we cannot rely on a main character classifier as we did in the script case, as we do not even have an approximate gold standard for main characters in the source video. Instead, we estimate the "main-characterness" $P^*(c \in main(M))$ (see Equation (4.3)) of a cluster as its share of the total number of interactions in the cluster network:

$$P^*(c \in main(M)) = \frac{\sum_{c' \in C_M} inter(c, c')}{\sum_{c_1, c_2 \in C_M \times C_M} inter(c_1, c_2)} \tag{4.27}$$

(cluster 1)



(cluster 2)

(cluster 3)

Figure 4.6: Examples of clusters retrieved for “The Silence of the Lambs”. Cluster 1 contains mainly faces that can be associated with Clarice Starling, the protagonist. Cluster 2 is highly correlated with Hannibal Lecter, her antagonistic “partner”. Cluster 3 is rather noisy, but seems to consist mainly of “male” faces.

4.7 Experimental Setup

Our experiments addressed two key aspects of the model presented here. Firstly, we are interested in the impact of the different objective terms pertaining to progression, diversity and importance (see Equations 4.2 ff.) on the visual scene extraction performance, in the uni-modal (script/video) as well as the multi-modal settings. Secondly, we want to verify that our scene extraction model indeed results in high quality, informative summaries for movies.

We have two ways of determining the compression rate m for the scene chain that our summarisation model should extract from the source video. One option is to set the compression rate relative to the *number of scenes*, e.g., for a movie consisting of 120 scenes and a compression rate of 10%, the model would extract exactly 12 scenes. The second option is to specify a certain *length* of the resulting video, either as a fixed

length, for example 10 minutes, or as a percentage of the total runtime of the source video, e.g., a 90 minute video at a compression rate of 10% would be summarised by extracting 9 minutes worth of scenes. For our human evaluations, we adopted the latter option throughout, by setting a compression rate relative to the source video runtime. We chose this setup to make summaries comparable. Firstly, two summaries produced for the same movie should be of equal length, and achieve the best summary possible within the allotted time. This is not guaranteed with a scene-based compression rate, as the length of scenes can vary dramatically. Secondly, for two different movies, a strict target summary time could disproportionally penalise longer movies. Intuitively, it seems likely that a 90 minute movie is more readily summarised in 10 minutes than a longer film of two hours length. We therefore opted to use a compression rate based on the movie's runtime in our experiments.

In order to evaluate our models, we performed two sets of experiments. The first round was conducted as an *intra-system* evaluation, aimed at identifying the best set of parameters in the script-based, movie-based, and multi-modal systems. We then conducted a second round of experiments, comparing the multi-modal system against comparable baseline extraction systems. In both cases, we relied on human judges to assess the quality of extracted summaries.

Two challenges we faced during evaluation were the availability of a large enough corpus of full-length feature films and, more severely, the lack of a gold standard for visual movie summaries. In order to have a dataset available, we acquired a total of 57 Hollywood movies. We selected 49 titles randomly from the *ScriptBase*-J dataset, and added the film versions of the original 8 test movies from Chapter 3. All films were then processed via the pipeline described in the previous section, detecting shots, identifying video scene boundaries through script-subtitle alignments, tracking faces throughout shots, and generating automatic face clusters and cluster networks. The shot detection threshold for histogram differences was set to 0.25, and we set the required confidence of the face detection model to 0.1 for a positive face detection. We set the initial number of clusters to be equal to the maximum number of parallel tracks identified in any shot, and increment it if clustering fails, e.g., if the linking constraints cannot be satisfied. For scene boundary identification, we aligned the movies' subtitles to their respective scripts, receiving time-stamps for scene beginnings and ends. All shots with timestamps contained within a so aligned scene were grouped together to obtain the corresponding visual scene.

As we could not easily overcome the lack of gold standard visual summaries, we

instead used the Wikipedia-aligned chains from Chapter 3. As these chains were obtained on a completely different modality and task, we only used them in a first step, to roughly estimate parameter settings for visual scene extraction. We then relied on human judges to assess summaries generated with various extraction models and parameter settings.

4.8 Results

4.8.1 Within-System Performance

We were first interested in the performance of our extraction model, *SceneSum*, in various modalities and parameter settings, in order to assess whether and how different parameters influence the summarisation quality, and how they behave in different modalities. To this end, we conducted a limited parameter estimation on Wikipedia aligned gold standard chains, and carried out a user-based study to assess the quality of summaries generated by scene extraction models based on script information alone (*SceneSum-S*), video based features (*SceneSum-V*), as well as a multi-modal system based on both sets of features (*SceneSum-M*).

Automatic Evaluation Before conducting the user study, we performed initial parameter tuning on 45 full movies, evaluated against Wikipedia plot aligned gold chains. We ran two rounds of evaluation, with parameters tuned on chains amounting to 10% of the movies' scenes, as well as parameters optimised for 10% of their runtime. In the scene setting, we found the best performing settings to be as follows: For *SceneSum-S*, we set $\lambda_D = 0.2$, and $\lambda_P = 1.0$. For *SceneSum-V*, the assignment is $\lambda_I = 1.0$. Finally, parameters for *SceneSum-M* are set to $\lambda_D = 0.4$, and $\lambda_P = 1.0$. Results for this setting are shown in Table 4.1, top. In the runtime case, the single best parameter setting for all three systems was $\lambda_P = 1.0$, with all other terms receiving zero weights. Results for the runtime-based extraction systems are shown in Table 4.1, bottom. Additionally in both scenarios, given the approximate nature of the gold standard chains, we also added systems using parameter settings in which *all* parameters are present, and chose those closest in performance to the "best" settings. In the scene-based setting, the parameter assignments were as follows: Parameters for *SceneSum-S* were set to $\lambda_D = 0.8$, $\lambda_I = 0.1$, $\lambda_P = 1.0$. For *SceneSum-M*, we assigned $\lambda_D = 0.1$, $\lambda_I = 1.0$, $\lambda_P = 0.1$. Finally, we set parameters for *SceneSum-M* to $\lambda_D = 1.0$, $\lambda_I = 0.1$,

10% of Scenes

	Best	All	Prog	Imp	Div
Script	21.9	18.8	12.7	5.5	20.0
Video	43.0	38.4	13.8	43.0	14.6
Multi-Modal	20.3	18.2	15.1	16.2	20.0

10% of Runtime

	Best	All	Prog	Imp	Div
Script	7.6	5.2	7.6	4.3	5.3
Video	6.2	5.0	6.2	5.1	3.8
Multi-Modal	6.2	4.6	6.2	4.4	4.1

Table 4.1: System performance of *SceneSum* modalities and parameter settings (45 movies). The top part shows systems optimised on number of scenes, the bottom shows system performance on movie runtime. We report F1-Scores on the Wikipedia plot aligned gold chains. "Best" are best parameter settings established for each system, "All" reflects closest performing setting including all parameters. Performances for systems using only progression, importance or diversity terms included for completeness.

$\lambda_P = 0.1$. When summarising based on movie runtime, we set model parameters as follows: *SceneSum-S* achieves best performance when setting $\lambda_D = 1.0$, $\lambda_I = 0.1$, and $\lambda_P = 0.1$. For *SceneSum-V*, the best assignment was $\lambda_D = 0.1$, $\lambda_I = 0.6$, and $\lambda_P = 1.0$. Finally, *SceneSum-M* is assigned $\lambda_D = 0.4$, $\lambda_I = 0.6$, and $\lambda_P = 1.0$. Results for all settings are shown in Table 4.1.

One fact that stands out from the automatic evaluation is the very strong performance of the video-only extraction in the scene-based setting. It achieves more than twice the F1 score of either script-based or multi-modal systems. When looking at the individual terms of the objective function, one can see that this performance stems solely from the *importance* term. While all systems seem to suffer on the gold standard chains, the video based system looses significantly more when factoring in *progression* and *diversity*. While this extreme behaviour is somewhat surprising, it again highlights the problem of the Wikipedia plot aligned evaluation chains, which we found to favour (a) long scenes, and (b) scenes containing a large number of characters. The *importance* term focuses specifically on these two aspects and, combined with the fact that script-video alignments are imperfect and some scene boundaries may be missed, the

video based system tends to generate summaries containing the 10% *longest* visual scenes.

When evaluating modalities and parameter settings on runtime, the achieved scores are very similar to each other. As no real gold standard for runtime-based evaluation is available, these systems were again evaluated against chains containing scenes that were aligned from movie scripts to Wikipedia plots, which are really designed for another task. This could explain the relative strength of the script-based system in this setting, as all the information available to it is specifically obtained from script scenes, to which the gold standard chains correspond. In all cases, we can observe that according to these gold standard chains, systems omitting one or more parameters entirely are achieving higher performances than those which take all terms into accounts.

Human Evaluation We are interested in how the summaries generated by our various models are perceived by their intended audience. We therefore conducted a user study to gather initial responses to the various extracted summaries. For this study, to make summaries comparable, we opted to test systems optimised on *runtime* rather than number of scenes. We selected four movies from our movie dataset, one each for the *action, drama, romance,* and *comedy* genres. We generated a total of 6 summaries for each movie, one for each modality and "best" and "all" parameter settings, respectively. The systems generated summaries for 10% of the movie runtime. We recruited 36 participants for the user study.

To ensure participants rated the summaries faithfully with respect to the movies, they were assigned one movie and one modality each. They then first watched the entire feature film, before seeing the two summaries generated for their assigned modalities. The modality and order of summaries was unknown to participants. We elicited responses with respect to the summaries' perceived quality, coherence and comprehensibility, the capturing of important points of the plot, as well as the overall movie coverage (beginning, middle, end). Each aspect was graded on a scale from 1 (bad) to 5 (very good). Participants then named their preferred of the two summaries they had been shown, and were given the opportunity to state what in particular they liked/disliked about either. Results of this human elicitation study are shown in Table 4.2.

The human evaluation seems to confirm the assumed bias of the previous gold-chain evaluation. In all cases, the higher performance of systems using the "best" parameter setting, i.e., taking into account only the *progression* term, over those using all objective terms are negated, or even reversed. This effect is strongest for *ScriptBase-*

Text-Based System

	Good Summary	Plot Points	Coherent	Coverage	Preferred
Best	2.58 / 2.0	2.67 / 2.5	2.5 / 2.5	2.58 / 2.0	6
All	2.58 / 3.0	2.92 / 2.5	2.67 / 2.5	3.0 / 3.0	6

Video-Based System

	Good Summary	Plot Points	Coherent	Coverage	Preferred
Best	2.0 / 3.0	2.08 / 2.0	2.33 / 2.0	1.75 / 2.0	2
All	3.25 / 2.5	3.33 / 3.0	3.25 / 3.0	4.08 / 4.0	10

Multi-Modal System

	Good Summary	Plot Points	Coherent	Coverage	Preferred
Best	2.58 / 3.0	2.75 / 3.0	2.92 / 3.0	2.75 / 2.5	6
All	2.58 / 3.0	2.75 / 2.5	2.92 / 3.0	2.67 / 3.0	6

Table 4.2: Results of human elicitation study, on within-system performance, for text-based (top), video-based (middle) and multi-modal systems. "Good Summary" refers to how adequate participants thought summaries were. "Plot Points" asked whether all relevant plot points were present in the summaries. Questions with respect to "Coherent" and "Coverage" assessed whether summaries exhibited an understandable structure, and whether they spanned the full movie. For all questions, participants graded the systems on a scale from 1 (bad) to 5 (very good). We report average (first number, before "/") and mean (second number, after "/") for each question.

V, for which the automatic evaluation indicated an advantage for the progression-only parameter setting. In stark contrast to the automatic evaluation, human judges overwhelmingly favoured the version of *ScriptBase-V* that optimised the full objective function. For the other systems, parameter settings perform equally well, but do negate the automatic results. However, differences between progression-based and full *SceneSum-S* and *SceneSum-M* systems in the results are not statistically significant, using two-tailed Mann-Whitney U test (Mann and Whitney, 1947). Note that this study evaluates different versions of the *same* model, and does not compare across models. Users saw two summaries of the same modality, and rated these systems relative to each other. Results are therefore biased within each modality, and not directly comparable across modalities[9].

[9]Even with that bias in mind, the same Mann-Whitney U test shows no significant differences between text-based and multi-modal settings.

4.8.2 Unseen Movie Evaluation

The previous experiment was aimed at assessing system performance as perceived by *knowledgeable* judges, that is, participants who were familiar with the movie that was being summarised. Parallel to the experiments of Chapter 3, we are also interested in assessing the *informativeness* of our extracted summaries. To this end, we ran an experiment similar in setup to the original script-based scene extraction system. We generated video summaries for the same 8 movies, using *SceneSum-S*, *SceneSum-V*, and *SceneSum-M*, at a compression rate of 10% of the movie's runtime.

This second experiment is complementary of the first human evaluation setup. Even though the video-only version of *SceneSum* seemed to perform better in the previous round of experiments, we still chose to evaluate all three system variations. We made this decision because we wanted to inspect whether despite of the worse performance when evaluated by knowledgeable judges, the multi-modal system is still able to produce *good* summaries. The second evaluation addresses this question in a more objective manner, as participants are not only asked whether they *think* the summaries were good, but also have to answer very specific questions, which are only really answerable if the important parts they address are covered by the summary.

In addition to summaries generated by the *SceneSum* systems, we also generated summaries for the same compression rate using two baselines. The first baseline selects random scenes from the video until the budget is satisfied. The second baseline selects scenes uniformly across the entire film. We then invited participants to watch and evaluate the summaries. We only accepted participants who had *not* seen the movies they were evaluating, and we elicited responses to the same questions pertaining to the movie's characters and plot points as in Chapter 3 (see also Appendix B for the full set of questions and expected answers). Figure 4.3 summarises the results of this user study.

SceneSum-M clearly outperforms both, the random and uniform selection baselines as well as the unimodal *SceneSum* systems on the tested movies as a whole. When comparing individual movies, *SceneSum-M* elicits the most correct answers from participants in the majority of cases (4 movies). *SceneSum-V* achieves the highest score for another two films, while the random and uniform models perform best for one movie each. Across movies, the performance of the random selection model is rather surprising. Leaving out the *SceneSum*, both the random and uniform systems outperform each other on an equal number of movies, eliciting better answers for four

films each. Looking at all movies, both systems perform comparably across all questions asked, however, the data suggests that the uniform system performs worse when it comes to the overall plot of the movie (question "What do you think this movie is about?"), yet outperforms the random baseline when it comes to answering specific questions pertaining to information contained in the movies. In any case, the *SceneSum* systems consistently outperform the baseline models. Within the three modality settings, the multimodal *SceneSum-M* model clearly performs better than the unimodal systems, scoring the highest marks in all but the main character identification task, where the video-based system takes a slight lead over the text-based extraction model. Overall, we observe that *SceneSum* in any setting is able to generate meaningful and informative summaries, for a wide variety of different movies. Furthermore, the multimodal variant that takes into account information obtained from both, the screenplay and the movies' visuals outperforms *SceneSum* models that only take into account either one of the modalities.

	Random	Uniform	SceneSum-S	SceneSum-V	SceneSum-M
Nightmare 3	45.56	34.07	42.96	40.37	**52.96**
Little Athens	34.35	31.29	41.20	41.48	**44.07**
Living in Oblivion	40.37	46.67	55.19	44.81	**66.29**
Mumford	35.06	**49.51**	48.89	44.94	35.31
One Eight Seven	31.02	27.22	45.65	**50.28**	43.24
Anniversary Party	**56.67**	26.30	42.10	31.60	29.51
We Own the Night	38.47	45.19	51.90	43.86	**53.70**
While She Was Out	55.19	58.89	43.33	**62.59**	59.63
All Questions	42.08	39.89	46.40	44.99	**48.09**
Five Questions	31.67	38.75	40.00	43.33	**45.42**
Plot Question	59.72	50.00	59.72	51.39	**63.89**
Characters Question	34.86	30.93	39.49	**40.26**	34.97

Table 4.3: Percentage of questions answered correctly in human evaluation.

4.9 Discussion

This chapter presented an adaptation of the original *SceneSum* model from Chapter 3 to the film domain. We extended the initial script-based scene extraction model

to include features obtained from the actual films. We showed how to exploit these features by adapting the *progression*, *diversity* and *importance* terms of the global objective function, to cover various aspects of the visual information, and generate visual summaries for feature films. In order to integrate both, script-based and video-based features into one system, we showed how to achieve script-video matching via script-subtitle alignments, coordinating script and visual scenes. Within-system evaluations showed that systems performed similar across different modality settings. Furthermore, human evaluation showed that contrary to indications from an automatic evaluation, incorporation of all partial objective terms is beneficial for the overall summary that is being extracted. A second round of human evaluations showed that our multi-modal *SceneSum-M* system produces summaries that are more informative than those generated by both random and uniform selection models, as well as the unimodal *SceneSum-S* and *SceneSum-V* systems, enabling participants to answer more questions that are highly relevant to the understanding of the movies.

One issue of the original model that is present and even heightened in the new, extended system is that of parameter optimisation. As observed for the script-based scenario, the dynamic program has to incorporate a very large number of variables and constraints into the solution, making exhaustive optimisation of local terms very time consuming. Instead, we again optimise the global objective function weights in order to achieve tractable parameter optimisation. Ideally, this optimisation would take into account each and every local objective and term. This could potentially be achieved by limiting the number of scene chains that are considered for extraction in the first place, either by manually engineered heuristics, or potentially even learned probabilities of certain scene combinations.

The adaptation to the film domain also brings some challenges with it. In particular, the identification of visual scenes and assignment to script scene boundaries via script-subtitle alignment is dependent on scenes appearing in the same order in both media. While this is the case in the vast majority of cases, some rearrangement can happen during production. Additionally, some film releases either drop entire scenes from their script, e.g., if they have to be cut out for time constraints, or add ad-hoc parts that were not scheduled in the original screenplay. Both poses challenges when integrating script information and information contained in the actual film. On the other hand, these challenges also provide for interesting avenues for future research, such as *deleted-scene detection*, or *reordering detection* between screenplays and motion pictures.

Another area that leaves room for improvement is the identification and cluster-

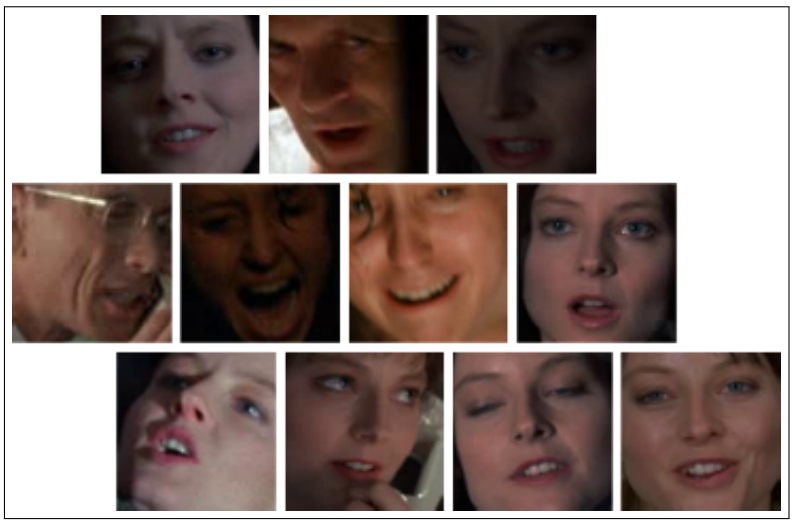


Figure 4.7: Example of an unexpected clustering result: The cluster seems to contain a disproportionally large number of open-mouthed faces, indicating the clustering algorithm picked up on this feature. Note that it also contains a high number of "Clarice" faces, which are thus missing from her main cluster.

ing of faces. While the state-of-the-art face detection and tracking approaches proved very reliable in our experiments, face clustering poses a significant challenge. While the constraint based approach employed in our setup is able to cluster, or not cluster, together faces of the same track and different tracks, respectively, the resulting face clusters still are rather impure. In some cases, the clusters do indeed correlate to specific characters in the film however, however, clusters often also contain other faces that just happen to look quite similar. There are also cases where the clustering resulted in collections of faces that seemed to rather correlate with other aspects such as *emotions* (e.g., open mouthed faces in shock, laughing or crying faces, see also Figure 4.7). We plan to further investigate this behaviour in particular, as it does have influence on the networks used during scene extraction, but also offers potential new ways to analyse character interactions, for example if different emotions can be reliably identified. We would also like to experiment on other methods of character identification in video, such as a mapping from faces to the movie script, where speakers are clearly marked.

In the future, we would like to address these challenges to further improve summarisation performance. Furthermore, we would like to extend *SceneSum* by additional modalities to complete the picture. In particular, after the script-based text and video-based film setting, the obvious next modality to include is the movie's audio. We expect that the summarisation extraction could greatly benefit from incorporating,

for example, features pertaining to a movie's score, sound effects, or spoken dialogue.

Chapter 5

A Joint Neural Network Architecture for Movie Content Analysis

The previous chapters presented models for the scene-based, *extractive* summarisation of movie scripts and feature films. Sections of the source scripts and movies were analysed with respect to their informativeness in a summary, and were aggregated as-is into a shortened version of the original material. This chapter presents a different approach to movie script summarisation, *abstracting* away from the original script, and representing its content as a high-level overview of a variety of the movie's aspects.

Drawing on recent advances in Neural Network Natural Language Generation, we motivate employing a jointly trained encoder-decoder model using feed-forward networks for attribute identification, and a Long Short-Term Memory mechanism for sentence realisation in order to generate informative texts which describe the underlying screenplays. Systems of this kind can be useful in a variety of tasks, from quickly providing a studio with useful information about certain aspects of a script before acquiring it, to the automatic generation of descriptive texts for new releases on movie recommendation sites or other service providers. Automatic and human evaluation show that the encoding of screenplays by attribute-value pairs is meaningful and the overviews provide informative and faithful descriptions of the movie's content.

5.1 Introduction

Movie content analysis is the task of analysing a film under different aspects such as its plot, genre, artistic style and so on, in order to gain an insight into the films general content. Within this setting, movie overview generation aims at automatically pro-

Mood:	Suspenseful, Captivating, Tense, Scary
Plot:	Serial Killer, Special Agents, Investigation, Mind Game, Psychopath, Crimes, Deadly, Law Enforcement, Mind and Soul, Rivalry
Genre:	Crime, Thriller
Style:	Strong Female Presence
Attitude:	Serious, Realistic
Place:	Maryland, USA, Virginia
Period:	20th Century, 90s
Based on:	Based on Book
Praise:	Award Winner, Blockbuster, Critically Acclaimed, Oscar Winner, Modern Classic, Prestigious Awards
Flag:	Brief Nudity, Sexual Content, Strong Violent Content

The Silence of the Lambs can be described as tense, captivating, and suspenseful. The plot revolves around special agents, mind games, and a psychopath. The main genres are thriller and crime. In terms of style, The Silence of the Lambs stars a strong female character. In approach, it is serious and realistic. It is located in Maryland and Virginia. The Silence of the Lambs takes place in the 1990s. It is based on a book. The movie has received attention for being a modern classic, an Oscar winner, and ablockbuster. Note that The Silence of the Lambs involves brief nudity and sexual content.

Figure 5.1: Jinni attributes, their values, and overview for “The Silence of the Lambs”. Attribute values which appear in the overview are underlined.

ducing a short natural language text that gives a general impression of the aspects of a movie. Potential applications of an automated overview generation system can be encountered at many stages of a movie’s life-cycle, from pre-production where a production company can use short, descriptive overviews to decide whether a potential new movie fits into the studio’s current direction, all the way to the consumer, especially on content streaming sites such as Netflix or Amazon Prime Video, for whom these content providers can easily generate overviews of new films that are coming to the service.

The attribute-value pairs and corresponding natural language overviews introduced with *ScriptBase*-J in Chapter 2, repeated in Figure 5.1 for convenience, represent the type of movie content analysis we would like to obtain automatically. Note how the

movie is labelled with tags for various *attributes*[1] (see the top half of Figure 5.1), which are then aggregated into a comprehensive overview (see the bottom half of Figure 5.1). While some of the presented attributes could not be possibly ascribed without information from external sources (e.g., *Praise*, or *Based on*), others could be inferred by watching the movie or reading the screenplay (e.g., *Genre, Plot, Flag, Mood, Place*). Furthermore, when organising the attributes into an overview, some *content selection* is necessary, as not all identified labels are actually used in the natural language text. For example, in Figure 5.1, the *mood* of the movie "The Silence of the Lambs" has been labelled as suspenseful, captivating, tense, and scary, but the corresponding sentence *"The Silence of the Lambs can be described as tense, captivating, and suspenseful."* omits the scary label.

This chapter presents a step toward automatic content analysis and summarisation of movie scripts by jointly modelling the tasks of movie attribute identification and overview generation. Specifically, we propose a novel neural network architecture which draws insights from encoder-decoder models recently proposed for machine translation (Bahdanau et al., 2015) and related sentence generation tasks (Wen et al., 2015; Mei et al., 2016; Lebret et al., 2016). Our model takes the screenplay as input and generates an overview for it. Rather than representing it as a sequence, we encode the screenplay via a set of content attributes and their values, viewing movie content analysis as a multi-label classification problem, since most movies naturally exhibit several labels per attribute (Snyder, 2005). For example, a movie can be simultaneously a thriller and a romance, and involve a heroic mission where people with supernatural abilities engage in the classic fight of good versus evil. Classification in this domain poses a significant challenge to data-driven methods due to the large number of labels (hundreds in the case of *Plot*) and the nature of the task which involves deep natural language understanding. Moreover, once the appropriate attributes have been identified it is not a priori obvious which ones should be included in the overview, as they typically range from high-level descriptions (e.g., Mind and Soul, Rivalry), to very specific ones (e.g., Special Agents, High School Life). We employ feed-forward multi-label classification networks (Zhang and Zhou, 2006; Kurata et al., 2016) for each attribute type (e.g., *Plot*, *Genre*) to encode the screenplay. Our decoder generates a movie overview using a Long Short-Term Memory network (LSTM; Hochreiter and Schmidhuber, 1997), a type of recurrent neural network with a more complex computational unit which is semantically conditioned Wen et al. (2015, 2016) on this attribute

[1]Throughout this chapter *attributes* are in italic font and their values in sans serif.

specific representation. Our model is trained end-to-end using screenplays and movie overviews as the supervision signal.

In the following we describe our neural network architecture in more detail and our efforts to create a dataset that consists of screenplays, Jinni-style attributes, and movie overviews. In both automatic and human-based evaluations our model outperforms competitive baselines and generates movie overviews which are well-received by human judges. To the best of our knowledge, this is the first work to automatically analyse and abstractively summarise the content of screenplays.

5.2 Related Work

Recent years have seen increased interest in the computational analysis of movie screenplays. Ye and Baldwin (2008) create animated storyboards using the action descriptions of movie scripts. Danescu-Niculescu-Mizil and Lee (2011) use screenplays to study the coordination of linguistic styles in dialog. Bamman et al. (2013) induce personas of film characters from movie plot summaries. Agarwal et al. (2014a) extract social networks from scripts. In subsequent work the same authors create *xkcd* movie narrative charts Agarwal et al. (2014b), and automate the Bechdel test (Agarwal et al., 2015) which is designed to assess the presence of women in movies. Chapter 3, Gorinski and Lapata (2015), summarise screenplays by selecting important scenes. Our work joins this line of research in an attempt to automatically induce information pertaining to a the content of a movie as its genre and plot elements.

In this chapter we propose a data-driven approach to movie overview generation based on neural networks. There has been a surge of interest recently in repurposing sequence transduction neural network architectures for NLP generation tasks such as machine translation (Sutskever et al., 2014), sentence compression (Chopra et al., 2016), and simplification (Zhang and Lapata, 2017). Central to these approaches is an encoder-decoder architecture modelled by recurrent neural networks. The encoder reads the source sequence into a list of continuous-space representations from which the decoder generates the target sequence. An attention mechanism (Bahdanau et al., 2015) is often used to locate the region of focus during decoding.

Previously proposed encoder-decoder architectures are not directly applicable to our task for at least two reasons: (a) the correspondence between screenplays and overviews is very loose, and (b) the screenplay is not strictly speaking a sequence (a screenplay is more like a book consisting of thousands of sentences), and cannot be

easily compressed into a vector-based representation from which the overview must be generated. Rather than attempting to decode the overview directly from the screenplay, we encode the later into attribute-value pairs which we then decode into overviews. We conceptualise the generation task as a joint problem of multi-label categorisation, where each screenplay is assigned to one or more categories, and content-sensitive natural language generation.

Many Machine Learning techniques have been proposed for building automatic text categorisation systems where a text is associated with multiple labels (see Sebastiani, 2002 and Dalal and Zaveri, 2011 for overviews), including neural networks (Belanger and McCallum, 2016; Kurata et al., 2016). We draw inspiration from multi-label classification in encoding screenplays as attribute-value pairs. Our encoder is essentially a feed-forward neural network, which however is able to capture label interactions which are important for our content analysis task. Although the use of templates has a long-standing tradition in text summarisation and information extraction (DeJong, 1982; Zhou and Hovy, 2004), we opt for on-the-fly generation, inspired by the recent success of LSTMs (Hochreiter and Schmidhuber, 1997) in text generation. Our decoder employs an enhanced LSTM architecture which directly maximises the probability of the overview given the screenplay's attribute values. Conditional LSTMs have been applied to various related tasks, including image description generation (Vinyals et al., 2015), the verbalisation of database records (Mei et al., 2016; Lebret et al., 2016), and the generation of dialogue acts (Wen et al., 2015, 2016).

5.3 Dataset

For our experiments, we make use of *ScriptBase*-J (Chapter 2). In particular, we utilise the Jinni attributes and overviews for the movies contained in the dataset. We split these 917 movies of *ScriptBase*-J into training, development and test sets, with 617, 200, and 100 instances, respectively. We concentrate on the six types of attributes shown in Table 5.1 whose values we hypothesise can be inferred from analysing the movie's screenplay. As mentioned earlier, attributes have values which are essentially labels/tags describing the movie's content, whereas overviews are short summaries giving a first impression of the movie. Table 5.1 provides an overview of the number of labels used in our experiments. Jinni contains a wealth of attribute values varying from nine for *Flag* to more than 400 for *Plot*. Additionally, value names for some attributes are synonyms or near-synonyms (e.g., Nudity and Brief Nudity for *Flag*). To make the

	Mood	Plot	Genre	Attitude	Place	Flag
Jinni	29	406	31	8	173	9
Frequent	19	101	31	8	53	9
Merged	19	101	31	8	24	6

Table 5.1: Movie attributes and their values used for overview generation. We used labels from six attribute sets. Where sets expressed a very large number of attributes, we excluded those attributes that occurred with low frequency. We also merged attributes that expressed the same or closely related concepts, such as violence and mild violence.

encoding task feasible, we reduced the set of attribute values to those that occurred most frequently (row "Frequent" in the table) and merged synonymous values into a common label (column "Merged").

5.4 Neural Overview Generation Architecture

We could approach the movie overview generation task using an attention-based encoder-decoder model (Bahdanau et al., 2015). The encoder would transform the input screenplay into a sequence of hidden states with an LSTM (Hochreiter and Schmidhuber, 1997) or another type of computational unit such as a gated recurrent unit (GRU; Cho et al., 2014a). The decoder would use another recurrent neural network to generate the overview one word at time, conditioning on all previously generated words and the representation of the input, while an attention mechanism would revisit the input sequence dynamically highlighting pieces of information relevant for the generation task. As mentioned earlier, viewing screenplays as a sequence of sentences is problematic both computationally and conceptually. Even if we used a hierarchical encoder (Tang et al., 2015; Yang et al., 2016) by first building representations of sentences and then aggregating those into a representation of a screenplay, it is doubtful whether a fixed length vector could encode the content of the movie in its entirety or whether the attention mechanism would effectively isolate the parts of the input relevant for generation.

We therefore propose an architecture that consists of two stacked neural network models for the tasks of movie attribute identification and overview generation. Figure 5.2 shows a schematic representation of our proposed architecture. Our architec-

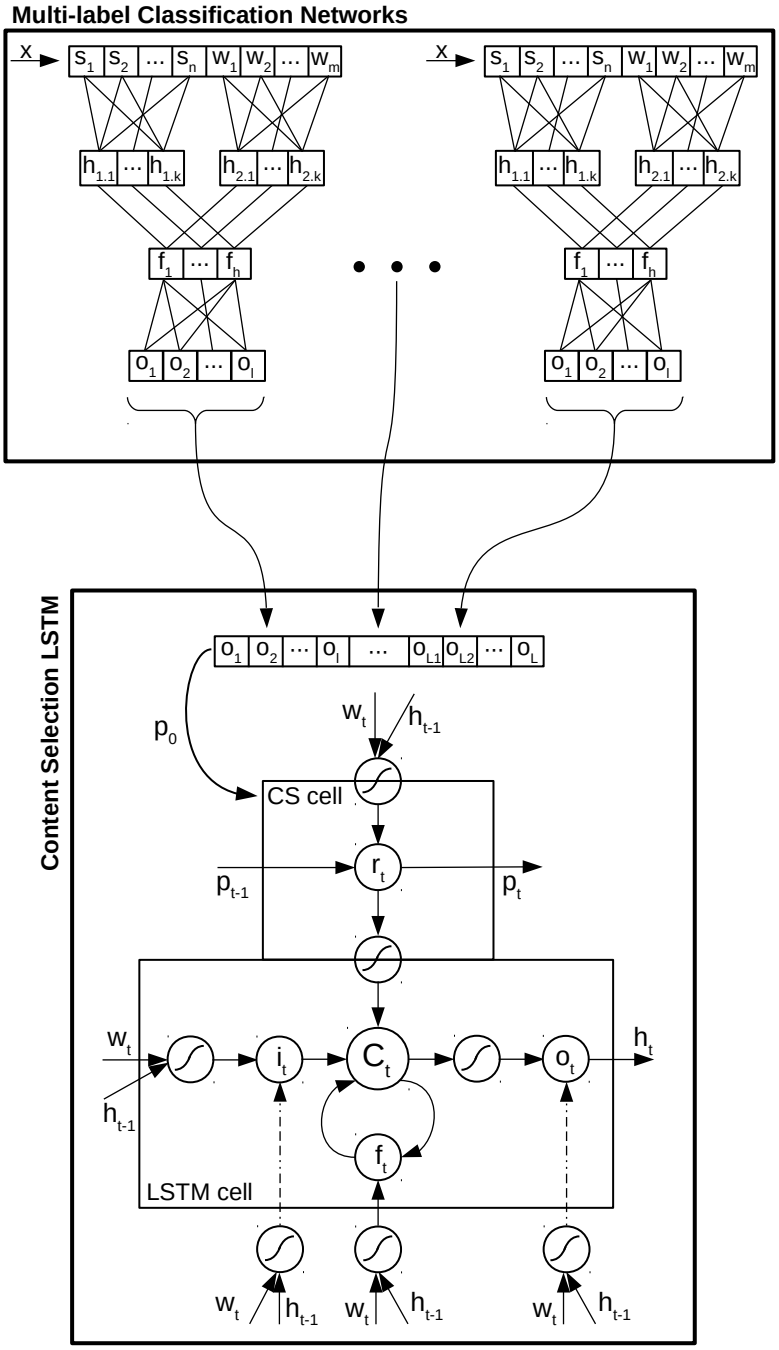


Figure 5.2: Movie overview generation architecture: given a vector x of input features representing a screenplay, we employ feed-forward multi-label classification networks to encode the movie into a content vector p_θ representing attribute labels; this encoding is fed into an LSTM with a content selection cell on top of the standard LSTM architecture.

ture uses simple feed-forward neural networks to impose some structure on the input screenplay by identifying the labels that most likely apply to it. We subsequently em-

ploy a semantically conditioned LSTM (Wen et al., 2015, 2016) to synthesise natural language overviews, utilising the label encoding to select the content for which to generate sentences. This architecture is advantageous for a number of reasons. Firstly, by imposing structure over the screenplays, the generation network is faced with a more compact and informative representation. This allows us to make use of a content selection LSTM similar to Wen et al. (2015, 2016), generating fluent and label-specific outputs. Secondly, it enables us to train the screenplay encoder (aka classification network) and the decoder jointly, in an end-to-end fashion. In the following we describe how overviews are generated via this joint model.

5.4.1 Multi-label Encoder

As shown in Figure 5.1 the overview highlights various aspects of the movie, essentially devoting a sentence to each attribute. This observation motivates us to encode the screenplay as a set of attributes (with their values) and then decode these into a sentence one by one. We treat attribute encoding as a multi-label classification problem: an attribute (e.g., *Genre* or *Plot*), will typically have multiple values (aka labels) which are suitable for the movie and should occur in the generated sentence. Furthermore, these labels naturally influence each other. For example, a movie whose *Genre* is Crime is also likely to be a Thriller while it is less likely to be a Parody. In traditional multi-label classification such interactions are either ignored (Read et al., 2011; Tsoumakas and Katakis, 2006; Godbole and Sarawagi, 2004; Zhang and Zhou, 2005), or represented by label combinations (Tsoumakas and Vlahavas, 2007; Read et al., 2008). A few approaches assume or impose an existing structure on the label space (Schwing and Urtasun, 2015; Chen et al., 2015; Huang et al., 2015; Jaderberg et al., 2014; Stoyanov et al., 2011; Hershey et al., 2014; Zheng et al., 2015), rather than letting the model discover label interactions.

We employ a neural network approach with the aim of abstracting the screenplay into a set of meaningful labels whose correlations are discovered automatically, during training. As shown on top of Figure 5.2, our encoder is a feed-forward neural network; using the sigmoid function for the network's output layer, where individual neurons represent the labels to be classified, the network can then give an estimate of the likelihood of which labels should be present. According to Equations (5.1)–(5.3), the input to the network is a feature vector x representing the screenplay (we discuss the specific features we use in the following section).

$$h_n = \sigma(W_n x_n) \tag{5.1}$$

$$f = h_1 \oplus h_2 \oplus \cdots \oplus h_k \tag{5.2}$$

$$O = \sigma(W_o f) \tag{5.3}$$

The input is split into k segments by feature type, and the feature segments are fed into k separate fully connected hidden layers. The hidden layer outputs are then combined using simple element-wise addition. The combined feature layer is used to compute an l-sized output layer, where l corresponds to the size of the classification label set. The final activation of the output units is obtained by applying the sigmoid function to the output layer. For our model, we employ three type of features representing the screenplay's lexical make up, its underlying character relations, and interactions.

Lexical Features An obvious feature class is the language of the movie. Comedies will be characterised by a different vocabulary compared to thrillers or historical drama. We thus represent each script as a vector of 7,500 dimensions corresponding to the most frequent words in the training corpus. Vector components were set to the words' tf-idf values. Words in scripts were further annotated with their sentiment values using the AFINN lexicon (Nielsen, 2010), a list of words scored with sentiment strength within the range $[-5, +5]$. We extracted several features based on these sentiment values such as the sentiment score of the entire movie, the number of scenes with positive/negative sentiment, the ratio of positive to negative scenes, and the minimum and maximum scene sentiment. From scene headings, we were also able to extrapolate the number of internal and external locations per script.

Graph-based Features Our graph-based features are similar to those described in Chapter 3, Gorinski and Lapata (2015). Specifically, we view screenplays as weighted, undirected graphs, where vertices correspond to movie characters and edges denote character-to-character interactions (essentially the number of times two characters talk to each other or are involved in a common action). From the graph we extract features corresponding to the number of main and supporting characters, which we identify by measuring their centrality in the movie network (e.g., the number of edges terminating in a given node). We also estimate character polarity by summing the sentiment of each character's utterances as well as the ratio of positive to negative characters in a given script.

Interaction-based Features We extract features based on how often any two characters interact, i.e., whether they are engaged in a conversation or in the same event (e.g., if a character kills another). We identify interactions as described in Chapter 3, Gorinski and Lapata (2015), and measure the number of interactions per scene and movie, the number of positive and negative interactions, and their ratio.

In order to better capture label interactions, we adapt a method of network initialisation recently introduced in Kurata et al. (2016). In this approach, instead of initialising the model's output weights W_o from a uniform distribution, the first p rows of the weight matrix are initialised according to patterns λ observed in the data. To this end, we initialise the nth row of W_o with initialisation pattern λ_n (Equation (5.4)), which is a vector corresponding to the nth label-assignment observed in the training data. The initialisation weight i for unit l of pattern λ_n is set to 0 if the corresponding label was not present in the given instance, or to the upper bound UB[2] (Glorot and Bengio, 2010) of the normalised initialisation weights of hidden layer h and output layer o, scaled by the number of times c the pattern occurs in the data (Equations (5.5), (5.6)). Figure 5.3 illustrates this initialisation procedure.

$$W_o^n = i(\lambda_n) \tag{5.4}$$

$$i(\lambda_n^l) = \begin{cases} \sqrt{c} \times UB & \text{if } \lambda_n^l = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

$$UB = \frac{\sqrt{6}}{\sqrt{|h| + |o|}} \tag{5.6}$$

In our setup, we limit the number of patterns p to the most frequently observed label assignments.

5.4.2 cs-LSTM Decoder

Our decoder generates a movie overview from the multi-label encoding described above. For this, we adapt the neural network architecture of Wen et al. (2015, 2016) which was originally designed for dialogue act generation (e.g., given the input *inform(type="hotel", count="182", dogsallowed="dontcare")*, the network outputs "*there are 182 hotels if you do not care whether dogs are allowed*"). The content selection network cs-LSTM decides which labels to talk about while generating the attribute describing sentence.

[2]We use the normalisation factor of $\sqrt{6}$ for UB, as suggested in Glorot and Bengio (2010)

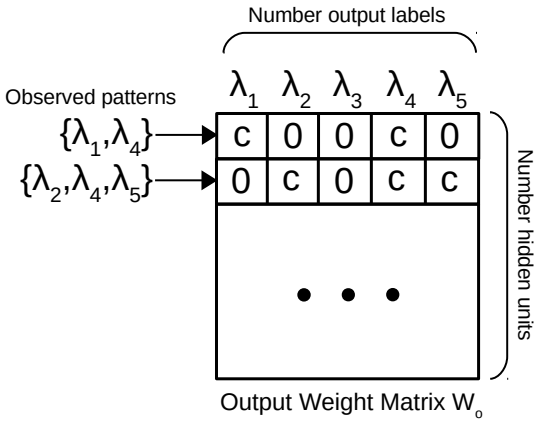


Figure 5.3: Outline of pattern initialisation in multi-label classification network.

As outlined in the lower part of Figure 5.2, a sigmoid control gate feeds a content vector, p_0, into a traditional LSTM cell to generate a corresponding natural language surface form. At each timestep t, the output word w_t is drawn from an output distribution conditioned on the previous hidden layer h_{t-1} *as well as* the previous content vector p_{t-1}. The content selection cell effectively acts as a *sentence planner*, retaining or omitting information from the original vector p_0 at every time step t to guide the sentence generating LSTM cell.

$$i_t = \sigma(W_{wi}w_t + W_{hi}h_{t-1}) \tag{5.7}$$

$$f_t = \sigma(W_{wf}w_t + W_{hf}h_{t-1}) \tag{5.8}$$

$$o_t = \sigma(W_{wo}w_t + W_{ho}h_{t-1}) \tag{5.9}$$

$$\hat{c}_t = \tanh(W_{wc}w_t + W_{hc}h_{t-1}) \tag{5.10}$$

$$r_t = \sigma(W_{wr}w_t + W_{hr}h_{t-1}) \tag{5.11}$$

$$p_t = r_t \odot p_{t-1} \tag{5.12}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t + \tanh(W_{pc}p_t) \tag{5.13}$$

In the original paper, the input p_0 to the cs-LSTM is a 1-hot representation of the information that should be included in the natural language output. In our setup, we relax this constraint such that each element of $p_0 \in [0,1]$, i.e., we directly use the output of the multi-label encoders described previously.

5.4.3 Training

The proposed architecture is trained jointly in an end-to-end fashion, minimising the same objective proposed by Wen et al. (2015), and given in equation (5.14):

$$F(\theta) = \sum_t y_t^T \log(\hat{y}_t) + ||p_T|| + \sum_{t=0}^{T-1} \eta\xi^{||p_{t+1}-p_t||} \tag{5.14}$$

where y_t and $\hat{y}_t$ are the observed and predicted word distributions over the training data, p_T is the content vector at the final time index T, p_0 is the initial content vector as given by the encoder network, and η, ξ are training constants. The second term in this objective penalises the network for generating output without realising all required labels, while the third term is used to deter the network from utilising more than one label at any given time step.

The model is trained on pairs of script features and sentences extracted from Jinni. To give a concrete example, a training instance for the *Plot* sentence from Figure 5.1 would consist of the features representing the movie's screenplay, and the *Plot* sentence of the overview *"The plot revolves around special agents, mind games, and a psychopath."*. The *Plot* multi-label network encodes the script into the content vector p_0, and the cs-LSTM learns which "labels" represented in p_0 to talk about while its training objective discourages to leave too many labels unmentioned. The observed output error is back-propagated through the cs-LSTM and the embedding network using stochastic gradient descent Bottou (1991) with decaying learning rate. We train for a maximum of 100 iterations over the training set with an initial learning rate of 0.5, decaying to 0.1 over 50 iterations.

5.5 Evaluation

In this section we report our evaluation experiments. We begin by assessing how good our encoder is at capturing screenplay content and then proceed to evaluated the generated reviews themselves. We also conduct human elicitation studies in order to assess how system output is perceived by users.

5.5.1 Automatic Evaluation: How Good is the Encoder?

In order to assess whether the feed-forward neural network encoders described in previous sections are able to induce structure over screenplays, we focus solely on the

	Mood	Plot	Genre	Attitude	Place	Flag	All
ZeroR	43.6	31.3	37.1	63.0	51.3	51.7	46.3
NB	51.2	36.7	52.4	68.5	49.7	49.3	51.3
DS	47.1	35.4	45.8	67.3	54.9	54.7	50.9
SVM	45.3	31.5	40.6	64.0	51.4	51.4	47.4
Lib	50.7	39.6	54.9	71.6	54.2	50.9	53.7
MLE	**58.4**	**43.9**	**55.3**	**76.5**	**58.6**	57.0	**58.3**

Table 5.2: Identification of movie attributes (average %F1 across 10 folds).

	Mood	Plot	Genre	Attitude	Place	Flag	All
ZeroR	43.4	32.8	37.8	61.2	48.2	48.8	45.4
Lib	48.0	38.9	54.6	69.0	46.1	46.4	50.5
MLE	**61.6**	**42.9**	**58.5**	**73.1**	**54.4**	**54.0**	**57.4**

Table 5.3: Attribute identification (%F1; test set).

top part of the architecture in Figure 5.2. Specifically, we trained stand-alone models for the six attributes shown in Table 5.1 on the gold data provided in the Jinni dataset. All networks used the same features introduced earlier and were initialised using the pattern-based method of Kurata et al. (2016). To better capture the fact that we are dealing with multi-label assignments, we used the non-standard global error function described in Zhang and Zhou (2006). Given the output vector $\hat{y}$ of the network for a given input x, as well as the true bag of label assignments y and its complement $\bar{y}$, the error observed for each instance is:

$$E = \frac{1}{|y||\bar{y}|} \sum_{(k,l)\in y\times\bar{y}} \exp(-(\hat{y}_k - \hat{y}_l)) \tag{5.15}$$

The networks were trained using stochastic gradient descent during back propagation, using the same method as for the full model.

We compared our multi-label encoders (MLE) against several baselines. These include assigning the most frequent attribute labels to each movie based on the attributes' mean distribution (ZeroR), Naive Bayes (NB), Decision Stump (DS), LibLinear (Lib; Fan et al., 2008) and Support Vector Machines (SVM; Chang and Lin, 2011). For each comparison system, we trained a binary classifier per attribute label using features identical to the ones used for the MLE.

As we have a fairly small amount of training data available, we performed cross-

Mood	T can be described as M_1 and M_2.
	The mood of T is M_1.
Plot	The plot centers around a P_1, P_2, and P_3.
	The plot revolves around P_1, P_2, and P_3.
Genre	The main genres are G_1, G_2, and G_2.
	T is M_1 and M_2 movie
Attitude	In approach, T is A_1.
	The pacing is A_1.
Place	The setting is L_1.
	It is located in L_1.
Flag	Note that it includes F_1, F_2, and F_3.
	Note that the movie involves F_1 and F_2.

Table 5.4: Template sentences extracted from Jinni. Variable T is filled by the movie's title, whereas M, G, P, A, L, F correspond to values for attributes *Mood*, *Genre*, *Plot*, *Attitude*, *Place*, and *Flag*, respectively.

validation for all systems. Table 5.2 shows F1 performance for MLE and comparison systems, averaged over 10 folds. As can be seen, MLE performs best, followed by LibLinear. Table 5.3 compares MLE, ZeroR, and Lib, the strongest baseline, on the test set using F1. Evaluation on the test set used the best parameters found for each system during cross-validation. As can be seen, MLE outperforms Lib on all attributes, and they are both superior to ZeroR by a large margin. F1 differences between MLE and LibLinear are significant ($p < 0.01$), using approximate randomisation testing (Noreen, 1989).

Overall, the results in Tables 5.2 and 5.3 indicate that the classification task is hard. This is especially true for *Plot* which has the largest number of labels. Nevertheless the multi-label encoders introduced here achieve good performance on their own, indicating that they are able to capture the content of the screenplay, albeit approximately.

5.5.2 Automatic Evaluation: How Good is the Decoder?

We next evaluate the performance of the jointly trained system which we call MORGAN as a shorthand for **M**ovie **O**ve**R**view **G**ener**A**tio**N** model. MORGAN was trained

on pairs of screenplays and their corresponding verbalisations in the Jinni dataset. Unfortunately, the dataset contains only 617 movies, i.e., there are 617 sentences for each attribute, which are hardly enough for neural network training. To alleviate this problem, we augmented the data as follows. We extracted sentence templates from the training set, examples of which are shown in Table 5.4. We replaced the title and attribute values with variables. We then used the templates to generate additional data for each movie by substituting attribute variables in template sentences with permutations of the movie's gold-standard labels. We thereby obtained a total of 31,000 training instances.

The full model was trained with a learning rate of 0.5, using a decay of 0.01 over 50 epochs, fixing it for subsequent epochs. Constants η and ξ in equation (5.14) were set to 10^{-4} and 100, respectively. At test time, we used screenplay features as input and generated one sentence per attribute. We arranged these into an overview following the ordering *Mood* $\gg$ *Plot* $\gg$ *Genre* $\gg$ *Attitude* $\gg$ *Place* $\gg$ *Flag* which is fixed and attested in all overviews in our dataset. An example overview produced by our system can be seen in Figure 5.5 (bottom) for "Burn After Reading", a movie from the test set.

We compared MORGAN against several systems: (1) A random baseline, selecting for each movie and sentence type a random sentence from the artificial training set; (2) a nearest-neighbour baseline (NN); (3) an attention-based LSTM (Bahdanau et al., 2015) trained on the complete set of sentences; and (4) a system with individually trained attention-based LSTMs, one per sentence type. The NN baseline uses the same screenplay features of as MORGAN, and uses cosine similarity to identify the closest matching script in the training data, whose corresponding sentences are used for the system's output. The attention LSTMs were trained on the same screenplay features as MORGAN, with the attention mechanism at each timestep t focusing on parts of the input. An example overview generated by this system is shown in Figure 5.5 (top). We evaluated system output with multi-reference BLEU[3] (Papineni et al., 2002), using sentences from the extended gold-standard as references. The document level BLEU scores achieved by MORGAN and the baseline systems are shown in Table 5.6 (column "BLEU").

As can be seen, MORGAN outperforms the attention based models, the nearest neighbour system, and the random baseline with respect to achieved BLEU scores. However, some baselines perform very close to MORGAN, in particular the *typAttn*

[3]We use python nltk's (http://www.nltk.org/) implementation of BLEU, and report the interpolation of BLEU1 through 4.

attnLSTM
The setting is Europe. It is a sci-fi, adventure, and adventure movie . Note that Burn after Reading includes violent content. Burn after Reading is set in the USA. It is located in the USA. It is set, at least in part, in the USA.
typed attnLSTM
The mood of Burn after Reading is bleak, touching, and feel good. The plot revolves around ambition motives, love and romance, and fall in love. It is an adventure and fantasy, and features thriller themes. In approach, Burn after Reading is serious. Burn after Reading is set, at least in part, in the USA. Note that Burn after Reading involves violent content.
Nearest Neighbour
The mood of Burn after Reading is stylized. The plot revolves around love and romance motives, buddies, and friends. Its main genres are romance, crime, and comedy. In approach, Burn after Reading is semi serious. It takes place, at least partly, in the USA. Note that Burn after Reading involves violent content, profanity, and sexual content.
MORGAN
The mood of Burn after Reading is feel good, exciting, and clever. The plot centers around ambition , special agents, and partners. Burn after Reading is a comedy movie. In approach, it is realistic and semi serious. It is set, at least in part, in the USA. Note that it involves profanity.

Table 5.5: Example of overviews generated from attention-based LSTM (top) and our model (bottom) for the movie “Burn after Reading”. Sentences are generated from screenplay features.

Models	BLEU	Coherence	Grammaticality
Random	38.0	2.42*	3.83
NN	40.4	3.45	3.93
Attn	23.0	2.93*	3.91
typAttn	37.9	3.20	3.80
MORGAN	**42.0**	**3.72**	**4.08**
Jinni	—	4.27	4.22

Table 5.6: BLUE scores and mean coherence and grammaticality ratings for movie overviews. * significantly different from MORGAN ($p < 0.05$). Best performing system shown in bold.

and nearest neighbour systems. This is potentially explained by the relatively simple structure of the Jinni overviews, and thus generated sentences. A good amount of overlap between gold and generated overviews can be achieved by generating the movie title and some non-attribute related words. The close performance of systems in this automatic evaluation is therefore of little surprise.

5.5.3 How are System Overviews Perceived by Humans?

In addition to evaluating system output automatically, we are also interested in how it is perceived by humans. To this end, we ran two judgement elicitation studies on Amazon Mechanical Turk.[4] Both experiments were conducted on 12 movies; these were the most-seen movies from the most popular genres in our dataset. In a pretest we asked 20 workers whether they had seen the movies in our test set and chose the three most popular ones from each of the genres Action, Comedy, Drama, and Romance.

In our first experiment Turkers were presented with an overview taken either from either the Jinni gold standard, MORGAN, or one of the baseline systems, and asked to rate overview's coherence (i.e., whether it was readily comprehensible or difficult to follow) on a scale from 1 (incoherent) to 5 (coherent). Subsequently, they were asked to rate the grammaticality of each overview sentence, again on a scale from 1 (ungrammatical) to 5 (grammaticality) and decide ("Yes", "No", "Unsure") whether it appropriately described aspects of the movie's content. We elicited five responses for

[4] https://www.mturk.com/

Model	Mood	Plot	Genre	Attitude	Place	Flag	All
Random	37.7	39.6	34.0	43.4	35.8	50.9	19.0*
NN	78.6	67.9	71.4	66.1	58.9	91.1	58.9*
Attn	38.2	38.2	38.2	41.8	51.0	34.5	40.0*
typAttn	60.0	60.0	53.3	57.8	**66.7**	64.4	40.0*
MORGAN	**89.5**	**73.7**	**80.7**	**71.9**	63.2	**89.5**	82.5
Jinni	91.1	89.3	92.9	82.1	67.9	75.0	91.1

Table 5.7: Proportion of sentences and overviews (All) which describe the movie accurately. * significantly different from MORGAN ($p < 0.05$). Best performing system per attribute is in bold.

each overview across three systems (Jinni, attnLSTM, and MORGAN) and 12 movies ($5 \times 3 \times 12$ overviews in total). Finally, participants had to answer a question relating to the movie's content, to make sure that they had actually seen the movie. We discarded responses with wrong answers to the content question. The questionnaire we used for this user study is also provided in Appendix D.

Table 5.6 summarises the results of our first judgement elicitation study. All systems perform well with regards to grammaticality. This is not surprising for Random and NN which do not perform any generation. Attn and typAttn also perform well, with MORGAN achieving highest scores for grammaticality amongst automatic systems. Grammaticality differences between the various systems in Table 5.6 and the Jinni gold standard are not statistically significant (using a one-way ANOVA with post-hoc Tukey HSD tests). Overviews generated by MORGAN are perceived as more coherent in relation to those generated by comparison systems, even though the model does not explicitly take coherence into account. MORGAN overviews are not significantly different in terms of coherence from Jinni, typAttn, and NN, but are significantly better than Random and Att.

Table 5.7 shows the percentage of sentences (per attribute and overall) which participants think describe the movie's content felicitously. MORGAN identifies most aspects of the movie successfully, in some cases close to (*Mood, Place*) or even better (*Flag*) than the original Jinni overview. MORGAN is significantly better compared to all other models (using a χ^2 test; see last column in Table 5.7) but not significantly worse than Jinni.

Model	1st	2nd	3rd	4th	5th	6th	AvgRank
Random	1.0	5.8	16.3	22.1	19.2	**35.6**	4.59
NN	5.8	19.2	**24.0**	23.1	15.4	12.5	3.60
Attn	3.8	13.5	20.2	**28.8**	16.3	17.3	3.92
typAttn	1.9	7.7	15.4	10.6	**33.6**	30.8	4.58
MORGAN	8.7	**42.3**	22.1	12.5	12.5	1.9	2.71
Jinni	**78.8**	11.5	1.9	2.9	2.9	1.9	1.45

Table 5.8: Relevance rankings (shown as proportions) given to overviews by human subjects. Most frequent rank per system and Jinni is in bold.

The perceived differences in the assessment of overviews by human readers can be explained in the way that content selection is addressed in the models. The attention-based models cannot succinctly capture the movie's content in order to render it into meaningful sentences, but have to rely on their attention to portions of the input to infer what a next "good" word to generate is. As also illustrated in Figure 5.5, although the generated sentences are grammatical on their own for most systems, the generated overviews lacks coherence, and do not always result in attributes that actually apply to the movie also appearing in the overview. The baseline models do not seem to reliably learn what type of information to focus on for the generation task. For MORGAN on the other hand, this problem is alleviated during the encoding step, which performs content distillation prior to generating overview sentences, and uses its LSTM decoder to organise this pre-selected content into overviews.

We were also interested in how human judges would rank summaries that were generated by different systems in direct comparison. We therefore conducted a second experiment in which participants were presented with six overviews for a movie, namely the gold overview from Jinni, the nearest neighbour overview for the movie, and overviews generated randomly, by the attention baselines, and by MORGAN. Participants were presented with all summaries at the same time, and asked to rank them in order of relevance, i.e., whether they express content relevant to the movie. Equal ranks were not allowed. Again, we obtained five responses for each movie. The questionnaire template for this human evaluation is also given in Appendix E. The results of this human ranking evaluation is shown in Table 5.8.

As can be seen, while Jinni is ranked first most of the time as expected, MORGAN

is ranked second in a majority of cases. The second ranked system, the nearest neighbour baseline, is ranked second only half as often as MORGAN. In a majority of the remaining ranks, *Morgan* is rated third much more often than any lower rank, with it being rated last in less than 2 percent of responses. system. We further converted the ranks to ratings on a scale of 1 to 6 (assigning ratings $6 \ldots 1$ to rank placements $1 \ldots 6$) and performed an ANOVA which showed that all systems are significantly ($p < 0.05$) worse than Jinni but MORGAN is significantly better than the comparison systems. These results suggest that the joint neural model presented here is well-suited for the task of automatic movie overview generation.

5.6 Discussion

In this chapter, we have presented a novel approach to automatic movie content analysis. We make use of the attributes and overviews provided by the *ScriptBase-J* corpus, and proposed an end-to-end model for movie overview generation via *multi-attribute* encoders and a *semantically conditioned* LSTM decoder. Experimental results show that our encoders are capable of distilling meaningful structures from the screenplay text. When applied to the overview generation task, our end-to-end model outperforms a standard attention-based LSTM approach. Human evaluation also indicates the overviews generated by our model are felicitous, informative, and are rated favourably by humans.

In the future, we would like to investigate how attribute-specific features can improve performance compared to our more general feature set which is invariant for each sentence type. It would also be possible to equip the model with a hierarchical decoder which generates a document instead of individual sentences. Although currently our model relies solely on textual information, it would be interesting to incorporate additional modalities such as video (Zhou et al., 2010) or audio (e.g., we expect comedies to be *visually* very different from thrillers, or romantic movies to have a different *score* from superhero movies). Finally, we would like to examine whether the content analysis presented here can extend to different types of fiction such as novels or short stories.

Chapter 6

Conclusions and Future Work

Automatic movie analysis and summarisation have seen many approaches and applications over the past years. This thesis has further advanced the field in several ways. With *ScriptBase* (Chapter 2), we have introduced a new resource to the community. We assembled a large corpus of movie scripts, accompanied by meta-data, Wikipedia plots, IMDB synopses and summaries, as well as log- and tag-lines. *ScriptBase* has been released, and is already seeing active use as a resource in NLP research. The corpus will be useful for various projects in many different areas of research. Tasks for which *ScriptBase* can be used include further research into movie analysis, for example discovering plot structures of films, or work in the social sciences, for example to explore how movies and their characters have changed over time. In Chapter 3 we presented a novel, character-character network based approach to the task of movie script summarisation. We showed how to synthesise script information into a network of the movie's characters, and how we can make use of the graph's properties to generate informative summaries. Chapter 4 showed how we adapt the *SceneSum* model to the film domain. We discussed how video-based features, in particular networks of characters represented through face clusters, can be obtained for use in the graph-based objective, and how they can eventually be combined with the original script-based features into a multi-modal model. Finally, in Chapter 5 we introduced MORGAN, a joint encoder-decoder neural network for the task of movie overview generation. MORGAN uses feed-forward encoders to pre-select aspects applicable to the movies, which are used by an LSTM-based sentence planner to generate natural language overviews.

The work we presented here has several implications for a variety of fields. In the context of Natural Language Processing, we have contributed work to the tasks of extractive text summarisation, as well as abstractive Natural Language Genera-

tion. For extractive summarisation, we have shown that a graph-based summarisation framework that is centred around characters and their relations is a viable option for analysing the content of movie scripts. Based on graph topography as well as properties of the characters, it is possible to efficiently and effectively identify important parts of a movie script, and organise such parts into coherent, informative summaries.

For Computer Vision, we have demonstrated the feasibility of a graph-based approach to video summarisation that focuses on interaction networks for faces clusters, which we interpret as characters. We have shown that for the task of summarisation in the movie domain, such networks are useful in helping to infer structural information about the movie, and in generating informative summaries. We have also demonstrated that the graph-based framework allows for straight-forward integration of video and script-based features into a multi-modal system that allows for the extraction of summaries that are highly informative and allow viewers to understand a large amount of movie content from the summaries alone.

In the area of Natural Language Generation, we have demonstrated that neural network models are a well suited for the task of generating highly abstractive analysis of movie scripts. Our results imply that the modelling of important information is feasibly performed on features directly inferred from scripts, through their language and structure. Furthermore, we have shown that the task can be addressed with jointly trained models, that do not have direct access to the correct labels that constitute the final overviews. This implies that overview generation for scripts can be addressed with minimal supervision, as all that is needed are source scripts, and target overviews, with no additional intermediary information.

In a more general case, our results imply that current Natural Language Processing, Computer Vision and Machine Learning technology can provide valuable tools for literary and film analysis. Researchers and other interest groups stand to benefit from computer aided approaches, such as graphical analysis of character relations in film, and processing tools and pipelines built on top of these approaches.

There are many avenues we want to take with future research. *ScriptBase* provides a rich resource for a great variety of tasks. In particular, we would like to use it to analyse in more depth the relationships between characters in movies, how they behave and interact with each other in snapshots of the movie, or over its entire length. Such work could stand on its own, but would also potentially be beneficial in future work on the script summarisation task. Deeper character analysis could yield networks that even closer model the true social structure of the movie, and in turn better inform scene

extraction. We would also like to explore a more fine-grained extraction of summaries, from a scene level down to a sub-scene, or even single actions and interactions of characters.

In the future we would like to refine *SceneSum* in the multi-modal video summarisation task. While human studies with the multi-modal system have been encouraging, there certainly remains work to be done. Future work in this direction includes further refinement of the video-based modelling of character networks, improvements to script-video alignment, as well as a more detailed exploration of modality interactions. Furthermore, the inclusion of additional modalities, such as sound in form of the film's score and spoken dialogue, is something we would like to address in future work. In the film domain, we would also like to explore how to automatically generate film trailers. While they are certainly related to video summaries, trailers pose a whole new class of challenges, maybe most notably the problem of spoiler prevention, but also the necessity of a more fine-grained extraction, from scenes down to single shots.

Finally, our neural movie overview generator also offers various avenues for future work. We would like to extend it to cover more of the attributes provided by the data set. For some, such as analysing a movie's score, we would like to also adapt a multi-modal version, capable of taking video or audio into account. We would also like to experiment with other encoders or decoders, and explore their impact on the overview generation process. On the encoder side, Structured Prediction Energy Networks (Belanger and McCallum, 2016) are of interest, as they are designed to directly model interdependencies of their output labels. On the decoder side, we would like to experiment with neural models other than LSTM such as Gated Recurrent Units (Cho et al., 2014b), which can be seen as a simplified LSTM, reducing processing costs. We would also like to explore the possibilities of taking hierarchical attribute information into account. Many labels in the dataset exhibit a "natural" structure (for example, a *parody* is a type of *comedy*, or *realistic* movie cannot at the same time be *fantastical*), and finding approaches to exploit these properties is a research task we would like to address. Another possible line of research is the model's adaptation to domains other than movie overviews. The model provides this possibility by adapting it to films and use video as input for the attribute classification stage, which should be relatively easy to achieve, since supervision requirements for the neural network architecture are minimal.

Appendix A

Movie Script Sources

This appendix provides the sources that were scraped when assembling the *ScriptBase* corpus in Chapter 2.

All scripts have been augmented with meta-data, summaries, synopses, tag lines, and log lines from `http://www.imdb.com`.
All scripts have been augmented with meta-data and overviews from `http://www.jinni.com`.
All scripts have been augmented with plot sections from `https://en.wikipedia.org`
`http://www.angelfire.com`
solarbabies

Scripts were crawled from the following sources:
`http://www.awesomefilm.com`
boys on the side; clue; come see the paradise; emma; lawn dogs; murderland; poetic justice; practical magic; pump up the volume; rent; the constant gardener; the doom generation; the forsaken; the jerk; this is spinal tap; working girl; chillfactor; deep end of the ocean; manhunt; pirscilla, queen of the desert; romy and michelle's high school reunion; suburbia; the parent trap; what women want; wonderland

`http://www.dailyscript.com`
austin powers: international man of mystery; below; blade ii; conspiracy theory; curse of the cat people; demolition man; domino; elf; entrapment; from russia with love; goldfinger; happy campers; hider in the house; jason x; mr. blandings builds his dream

house; on the waterfront; pi; rambo: first blood part ii; star trek: the motion picture; the fisher king; the life and death of colonel blimp; the ploughman's lunch; the third man; the wedding date; virtuosity; beetle juice; blair witch ii; flight plan; good fellas; hero; o brother, where art thou?; the adventures of ford fairlane; the dragons of krull; the jolson story

`http://www.horrorlair.com`

a nightmare on elm street 2: freddy's revenge; a nightmare on elm street 3: dream warriors; freaks; friday the 13th part iii; friday the 13th: the final chapter; i am legend; i know what you did last summer; jaws 2; jaws 3-d; jaws: the revenge; m; natural born killers; nosferatu; phantasm; plan 9 from outer space; stigmata; stranglehold; the craft; the faculty; the house next door; the morgue; the night of the hunter; the silence of the lambs; urban legend; waxwork; a nightmare on elm street 4; a nightmare on elm street 5; a nightmare on elm street 6; a nightmare on elm street 7; amityville horror; bram stoker's dracula; bucket of blood; company of wolves; count dracula; dark city; dracula (1931); dracula (1979); friday the 13th part iv: jason lives; halloween 2; halloween 4; halloween 8: resurrection; halloween h20; rebecca; shock treatment; stone tape; the horror of dracula; the old dark house; the running man; willard

`http://www.imsdb.com`

(500) days of summer; 10 things i hate about you; 12; 12 and holding; 12 monkeys; 17 again; 2012; 25th hour; 30 minutes or less; 44 inch chest; 48 hrs.; 50/50; 8mm; a dry white season; a few good men; a perfect world; a serious man; a walk to remember; above the law; absolute power; adaptation; after.life; agnes of god; air force one; airplane ii: the sequel; airplane!; alien; alien vs. predator; aliens; all about steve; all the president's men; almost famous; alone in the dark; amadeus; amelia; american beauty; american gangster; american graffiti; american history x; american madness; american pie; american splendor; an american werewolf in london; an education; analyze that; analyze this; anastasia; angel eyes; angels & demons; annie hall; anonymous; antitrust; antz; apocalypse now; april fool's day; apt pupil; arbitrage; arcade; arctic blue; argo; armageddon; army of darkness; autumn in new york; avatar; awakenings; babel; bachelor party; bad boys; bad day at black rock; bad dreams; bad lieutenant; bad santa; bad teacher; bamboozled; barry lyndon; barton fink; basic; basic instinct; basquiat; batman; battle: los angeles; beasts of the southern wild; beavis and butthead do america; beginners; being human; being john malkovich; beloved; benny &

joon; big; big fish; birthday girl; black snake moan; black swan; blade; blade runner; blade: trinity; blood and wine; blood simple; blow; blue valentine; body heat; bones; bonnie and clyde; bottle rocket; bound; breakdown; brick; bridesmaids; bringing out the dead; broadcast news; broken arrow; broken embraces; bruce almighty; buried; burn after reading; capote; cars 2; case 39; casino; cast away; catch me if you can; catwoman; cecil b. demented; cedar rapids; cellular; changeling; charade; charlie's angels; chasing amy; chasing sleep; cherry falls; chinatown; cinema paradiso; cirque du freak: the vampire's assistant; clash of the titans; clerks; cliffhanger; clueless; cobb; code of silence; cold mountain; collateral; collateral damage; colombiana; conan the barbarian; confessions of a dangerous mind; confidence; constantine; coraline; coriolanus; cowboys & aliens; crank; crime spree; crouching tiger, hidden dragon; croupier; cruel intentions; cube; dances with wolves; dark star; darkman; date night; days of heaven; deception; deep cover; defiance; devil in a blue dress; die hard; die hard 2; diner; disturbia; django unchained; do the right thing; dogma; donnie brasco; double indemnity; drag me to hell; dragonslayer; drive; drive angry; drop dead gorgeous; dune; e.t. the extra-terrestrial; eagle eye; eastern promises; easy a; ed wood; edward scissorhands; eight legged freaks; election; elizabeth: the golden age; enemy of the state; enough; erik the viking; erin brockovich; escape from l.a.; escape from new york; eternal sunshine of the spotless mind; even cowgirls get the blues; event horizon; excalibur; existenz; extract; face/off; fair game; fantastic four; fantastic mr. fox; fast times at ridgemont high; fatal instinct; fear and loathing in las vegas; feast; ferris bueller's day off; field of dreams; final destination; final destination 2; finding nemo; five easy pieces; flash gordon; flight; forrest gump; four rooms; frances; frankenstein; freaked; frequency; friday the 13th part viii: jason takes manhattan; from dusk till dawn; from here to eternity; frozen river; funny people; g.i. jane; g.i. joe: the rise of cobra; game 6; gamer; gandhi; gang related; gangs of new york; gattaca; get carter; get low; get shorty; ghost; ghost rider; ghost ship; ghost world; ghostbusters; ghostbusters ii; ginger snaps; gladiator; glengarry glen ross; go; gods and monsters; gone in 60 seconds; good will hunting; gothika; gran torino; grand hotel; grand theft parsons; gremlins; grosse pointe blank; groundhog day; hackers; hall pass; halloween: the curse of michael myers; hancock; hannah and her sisters; hannibal; hard rain; he's just not that into you; heavenly creatures; heist; hellboy; hellraiser: deader; henry fool; henry's crime; hesher; high fidelity; highlander: endgame; his girl friday; hollow man; honeydripper; horrible bosses; hostage; hot tub time machine; hotel rwanda; house of 1000 corpses; how to lose friends & alienate people; how to train your dragon; hudson

hawk; human nature; i am number four; i love you phillip morris; i, robot; in the bedroom; in the loop; indiana jones and the last crusade; indiana jones and the temple of doom; inglourious basterds; insidious; interview with the vampire: the vampire chronicles; into the wild; intolerable cruelty; inventing the abbotts; invictus; it happened one night; it's complicated; jackie brown; jacob's ladder; jane eyre; jay and silent bob strike back; jennifer's body; jerry maguire; jimmy and judy; john q; judge dredd; juno; jurassic park iii; kate & leopold; kids; killing zoe; klute; kramer vs. kramer; kundun; kung fu panda; l'avventura; l.a. confidential; labyrinth; lake placid; land of the dead; larry crowne; last chance harvey; last tango in paris; legally blonde; legend; legion; leviathan; liar liar; life; life as a house; life of pi; light sleeper; limitless; lincoln; little athens; little nicky; lock, stock and two smoking barrels; looper; lord of illusions; lord of war; lost highway; lost in translation; machete; machine gun preacher; malibu's most wanted; man on fire; man on the moon; man trouble; manhunter; margin call; margot at the wedding; marley & me; martha marcy may marlene; marty; mary poppins; mean streets; meet joe black; meet john doe; miami vice; midnight cowboy; midnight express; midnight in paris; mighty morphin power rangers: the movie; milk; miller's crossing; mini's first time; minority report; mirrors; misery; mission to mars; moneyball; monkeybone; moon; moonrise kingdom; moonstruck; mr. brooks; mr. deeds goes to town; mrs. brown; mulan; mulholland drive; music of the heart; my best friend's wedding; my girl; my mother dreams the satan's disciples in new york; my week with marilyn; mystery men; napoleon dynamite; new york minute; newsies; next; next friday; nightbreed; nine; ninja assassin; ninotchka; no strings attached; notting hill; oblivion; observe and report; obsessed; ocean's eleven; ocean's twelve; office space; one flew over the cuckoo's nest; only god forgives; ordinary people; orgy of the dead; orphan; pandorum; panic room; paranorman; paul; pearl harbor; peeping tom; peggy sue got married; perfect creature; pet sematary; philadelphia; phone booth; pineapple express; pirates of the caribbean: dead man's chest; pitch black; poltergeist; precious; predator; pretty woman; priest; prom night; prometheus; public enemies; punch-drunk love; purple rain; queen of the damned; rachel getting married; raging bull; raising arizona; rambling rose; real genius; rebel without a cause; red planet; red riding hood; reindeer games; repo man; reservoir dogs; revolutionary road; rise of the planet of the apes; rko 281; robin hood: prince of thieves; rocknrolla; romeo & juliet; roughshod; runaway bride; rush hour; rush hour 2; rust and bone; s. darko; saving private ryan; scarface; schindler's list; scott pilgrim vs. the world; scream; scream 2; scream 3; semi-pro; sense and sensibility; serenity; serial mom; sex and the

city; sex, lies and videotape; sexual life; shakespeare in love; shallow grave; shame; shampoo; she's out of my league; sherlock holmes; shifty; shivers; shrek; shrek the third; sideways; signs; silver linings playbook; simone; sister act; six degrees of separation; sleepless in seattle; slumdog millionaire; smashed; smokin' aces; snatch; snow falling on cedars; snow white and the huntsman; someone to watch over me; something's gotta give; source code; spare me; st. elmo's fire; star trek; star trek: nemesis; star wars episode i: the phantom menace; star wars episode ii: attack of the clones; star wars episode iii: revenge of the sith; starman; state and main; station west; stepmom; stir of echoes; storytelling; strangers on a train; sugar; sunshine cleaning; superbad; supergirl; surrogates; suspect zero; sweet smell of success; swingers; swordfish; synecdoche, new york; taking lives; taking sides; tall in the saddle; tamara drewe; taxi driver; ted; terminator salvation; the abyss; the addams family; the adjustment bureau; the adventures of buckaroo banzai across the 8th dimension; the american; the american president; the apartment; the artist; the assignment; the avengers; the bachelor party; the back-up plan; the battle of algiers; the battle of shaker heights; the beach; the believer; the best exotic marigold hotel; the big blue; the big lebowski; the big white; the black dahlia; the blind side; the book of eli; the boondock saints; the bounty hunter; the bourne identity; the bourne supremacy; the bourne ultimatum; the box; the breakfast club; the brothers bloom; the butterfly effect; the cell; the change-up; the cider house rules; the cincinnati kid; the cooler; the crow: city of angels; the crow: salvation; the crying game; the curious case of benjamin button; the damned united; the dark knight rises; the day the clown cried; the deer hunter; the departed; the descendants; the elephant man; the english patient; the evil dead; the family man; the fifth element; the fighter; the flintstones; the french connection; the fugitive; the game; the ghost and the darkness; the girl with the dragon tattoo; the godfather; the godfather part ii; the godfather part iii; the good girl; the grapes of wrath; the green mile; the grifters; the grudge; the hangover; the haunting; the hebrew hammer; the help; the hitchhiker's guide to the galaxy; the horse whisperer; the hospital; the hudsucker proxy; the ice storm; the ides of march; the incredibles; the insider; the invention of lying; the iron lady; the island; the italian job; the jacket; the kids are all right; the king of comedy; the king's speech; the kingdom; the last boy scout; the last flight; the last samurai; the last station; the life of david gale; the limey; the lincoln lawyer; the long kiss goodnight; the lord of the rings: the return of the king; the lord of the rings: the two towers; the losers; the man who knew too much; the man who wasn't there; the matrix; the mechanic; the men who stare at goats; the neverending story; the next

three days; the nightmare before christmas; the nines; the pacifier; the passion of joan of arc; the patriot; the perks of being a wallflower; the pianist; the piano; the postman; the power of one; the princess bride; the private life of sherlock holmes; the program; the prophecy; the queen; the rage: carrie 2; the reader; the relic; the replacements; the rescuers down under; the rock; the rocky horror picture show; the roommate; the ruins; the saint; the salton sea; the sessions; the seventh seal; the shawshank redemption; the shining; the shipping news; the siege; the sixth sense; the thing; the things my father never taught me; the three musketeers; the tourist; the ugly truth; the usual suspects; the verdict; the village; the way back; the whistleblower; the white ribbon; the wild bunch; the wizard of oz; the woodsman; the world is not enough; they; this boy's life; this is 40; thor; three kings; three men and a baby; thunderheart; tin cup; tin men; tinker tailor soldier spy; titanic; tmnt; to sleep with anger; tombstone; tomorrow never dies; traffic; trainspotting; tremors; tron; true lies; true romance; twilight; twin peaks: fire walk with me; twins; two for the money; unbreakable; under fire; unknown; up; up in the air; v for vendetta; valkyrie; vanilla sky; very bad things; wag the dog; wall street; wall-e; wanted; war horse; warm springs; warrior; watchmen; water for elephants; we own the night; what about bob?; what lies beneath; while she was out; white christmas; wild at heart; wild hogs; wild things: diamonds in the rough; wild wild west; willow; win win; withnail and i; witness; wonder boys; x-men origins: wolverine; xxx; year one; yes man; you can count on me; you've got mail; outh in revolt; zero dark thirty; zerophilia; 13 days; 1492: conquest of paradise; 15 minutes; 187; 9; a nightmare on elm street; ace ventura: pet detective; after school special; aladdin; ali; alien nation; all about eve; american shaolin; amour; anna karenina; arthur; austin powers 2: the spy who shagged me; badlands; bean; being there; black rain; bodyguard; bonfire of the vanities; boondock saints 2: all saints day; buffy, the vampire slayer; burlesque; burning annie; cable guy; carrie; celeste & jesse forever; chaos; city of joy; commando; copycat; cradle 2 the grave; crash; crazy, stupid, love; dawn of the dead; day of the dead; death at a funeral; death to smoochy; deep rising; detroit rock city; devil's advocate; duck soup; edtv; evil dead 2; fracture; freddy vs. jason; friday the 13th; fright night; fright night (1985); frozen; godzilla; gremlins 2; hanna; happy birthday, wanda june; hard to kill; harold and kumar go to white castle; heat; heavy metal; hellboy 2: the golden army; hellraiser 3: hell on earth; hellraiser: hellseeker; hitchcock; indiana jones and the raiders of the lost ark; indiana jones iv; insomnia; jennifer eight; kill bill volume 1; kill bill volume 2; king kong; law abiding citizen; les miserables; logan's run; lone star; lost horizon; love and basketball; mad max 2: the road warrior; malcolm

x; margaret; max payne; megamind; monte carlo; nashville; never been kissed; nick of time; pariah; pet sematary ii; petulia; pirates of the caribbean; platinum blonde; point break; pokemon: mewtwo returns; pride and prejudice; psycho; rear window; remember me; ringu; ronin; save the last dance; saw; se7en; slither; solaris; soldier; south park: bigger, longer, uncut; spartan; speed racer; star trek: first contact; star trek: generations; sugar and spice; sunset blvd.; super 8; sweeney todd: the demon barber of fleet street; the anniversary party; the blast from the past; the chronicles of narnia: the lion, the witch and the wardrobe; the crow; the day the earth stood still; the debt; the distinguished gentleman; the four feathers; the getaway; the hills have eyes; the imaginarium of doctor parnassus; the informant; the ladykillers; the last of the mohicans; the little mermaid; the lord of the rings: the fellowship of the ring; the lost world: jurassic park:; the majestic; the man in the iron mask; the manchurian candidate; the master; the matrix reloaded; the miracle worker; the other boleyn girl; the producers; the proposal; the road; the sandlot kids; the searchers; the sting; the stuntman; the sweet hereafter; the taking of pelham one two three; the visitor; the wrestler; the x-files: fight the future; ticker; timber falls; top gun; transformers: the movie; tristan and isolde; tron: legacy; tropic thunder; true grit; twilight: new moon; walking tall; white squall; whiteout; who framed roger rabbit?; who's your daddy; wind chill

`http://www.scenebyscene.net`

star wars episode iv: a new hope; star wars episode vi: return of the jedi

`http://www.scifiscripts.com`

2001: a space odyssey; back to the future; escape from the planet of the apes; fantastic voyage; galaxy quest; highlander; little monsters; mimic; mission: impossible; spaceballs; star trek ii: the wrath of khan; star trek v: the final frontier; star trek vi: the undiscovered country; starship troopers; superman ii; superman iii; superman iv: the quest for peace; the 13th floor; the terminator; the witching hour; things to come; thx 1138; troops; war of the worlds; back to the future iii; battlestar galactica; dr. strangelove: or, how i learned to stop worrying and love the bomb; highlander 3; independence day; la jetee; superman: the motion picture; the mummy; west world

`http://sfy.ru`

a night at the roxbury; a.i. artificial intelligence; affliction; alien 3; alien: resurrection; american psycho; an american tragedy; apollo 13; as good as it gets; assassins; at first

sight; backdraft; batman & robin; batman returns; blue velvet; bodies, rest & motion; boiler room; boogie nights; brazil; bull durham; citizen kane; cool hand luke; cross of iron; dead poets society; dog day afternoon; donnie darko; dumb and dumber; el mariachi; eyes wide shut; fargo; fight club; harold and maude; heathers; hellbound: hellraiser ii; hellraiser; i am sam; i still know what you did last summer; i walked with a zombie; i'll do anything; isle of the dead; jaws; jfk; joe versus the volcano; jurassic park; kafka; kalifornia; leaving las vegas; lethal weapon; living in oblivion; lost in america; lost in space; lost souls; magnolia; major league; mash; melvin and howard; memento; midnight run; mission: impossible ii; mobsters; monty python and the holy grail; monty python live at the hollywood bowl; monty python's the meaning of life; mr. smith goes to washington; mumford; my own private idaho; network; nixon; nothing but a man; nurse betty; passenger 57; platoon; pleasantville; rabid; rapture; resident evil; return to me; ride the high country; rushmore; scary movie; scary movie 2; shine; silverado; sleepy hollow; sling blade; smoke; so i married an axe murderer; sounder; spanglish; sphere; stalag 17; star wars episode v: the empire strikes back; strange days; the african queen; the birds; the blair witch project; the body snatcher; the bridges of madison county; the corruptor; the devil and daniel webster; the doors; the exorcist; the fabulous baker boys; the goonies; the graduate; the hustler; the jackie robinson story; the last temptation of christ; the leopard man; the lost boys; the lost weekend; the messenger: the story of joan of arc; the ninth gate; the omega man; the seventh victim; the straight story; the talented mr. ripley; the thin man; the thing called love; the truman show; thelma & louise; there's something about mary; toy story; u turn; vertigo; viridiana; when harry met sally...; white angel; wild things; x-men; 84 charlie mopic; all the king's men; body of evidence; braveheart; cat people; contact; father of the bride; fletch; full metal jacket; halloween; happiness; hardcore; harvey; house on haunted hill; innerspace; it's a wonderful life; leon; life of brian; made for each other; men in black; metro; one good turn; only you; out of sight; pulp fiction; rocky; salt of the earth; silver bullet; some like it hot; spider-man; stagecoach; terminator 2: judgement day; the age of innocence; the fly; the great train robbery; the jazz singer; the mask; the scarlet letter; the swimmer; tomb raider; total recall; true believer; unforgiven; when a stranger calls; planet of the apes; planet of the apes; the time machine; the time machine

`http://leonscripts.tripod.com`

atomic submarine; invaders from mars; invaders from mars

`http://www.pages.drexel.edu`

children of men; dave; idiocracy; quills; requiem for a dream; the dark knight; dexter; pushing daisies: pie-lette

`http://www.weeklyscript.com`

a hard day's night; an american werewolf in paris; an officer and a gentleman; apocalypse now redux; boy who never slept; casablanca; hope and glory; little miss sunshine; no country for old men; panther; portrait of jennie; smokey and the bandit; stolen summer; the 40-year-old virgin; the public eye; the thing from another world; the wild one; young frankenstein; zulu dawn; batman begins; naked city

`http://www.wingkong.net`

big trouble in little china

Appendix B

Movie Summarisation Questionnaires

This appendix provides the questions (Q) given to participants of the human movie summarisation evaluation study of Chapter 3.5.2, and the second human evaluation study of Chapter 4.8.2, along with examples of expected answers (A).

In all cases, users where asked the following questions:
Q: *On a scale from 1 to 5 (where 1 is worst and 5 is best), how well do you think you understood the overall plot of the movie, based on the scenes selected by the computer program?*

Q: *On a scale from 1 to 5 (where 1 is not at all, and 5 is completely), how well do you think the automatic summary covered the entirety of the movie (beginning/middle/end)?*

Q: *OPTIONAL: Is there anything that you think was particularly good/bad about the automatic summary?*

For the movie "187", the following questions were asked:
Q: *Who are the main characters of the movie?*
A: Trevor (Samuel L. Jackson), Ellen (Kelly Rowan), Cesar (Clifton Collins Jr.), Rita (Karina Arroyave), Benny (Lobo Sebastian), Larry (Jack Kehler), Principal Garcia (Tony Plana), Stevie (Jonah Rooney)

Q: *Please give a short overview of what you think the movie is about.*
187 is about Trevor, a traumatised teacher from New York, who moves to LA and takes over a class full of problem kids. He gets head-to-head with some of the students, especially some members of a brutal local gang. He learns that the gang threatened students and teachers, in particular Ellen who Trevor likes. Trevor takes it upon himself to fight

the established system of fear, with legal and illegal approaches.

Q:*Why does Trevor leave New York and where does he move to?*
A: Trevor leaves because he was attacked and severely wounded by a student in his old school. He moves to LA.

Q:*What is KOS, who is their leader, and why is he attending high school?*
A: KOS is a local gang "Kappin' Off Suckers", Benito is their leader, and he attends high school as part of a probation program.

Q: *What happened to Cesar's finger, how did he eventually die?*
Cesar's finger gets cut off, and has a warning tattooed to it. He shoots himself in the head in a game of Russian Roulette.

Q:*Who killed Benny and how does Ellen find out?*
A: Trevor killed Benny. Ellen suspects it after she sees Trevor with Benny's rosary beads.

Q: *Who is Rita and what becomes of her?*
A: Rita is one of the problem students Trevor teaches in LA. She graduates, and even holds the graduation speech.

For the movie "Little Athens", the following questions were asked: Q: *Who are the main characters of the movie?*
A: Jimmy (John Patrick Amedori), Carlos (Michael Pea), Allison (Rachel Miner), Heather (Erica Leerhsen), Corey (DJ Qualls), Pedro (Jorge Garcia), Jessica (Jill Ritchie), Aaron (Kenny Morrison)

Q: *Please give a short overview of what you think the movie is about.*
A: Jimmy deals drugs and has money problems, steals Kwon's drugs from dead dealer and tries to sell them at party. Heather thinks her boyfriend Derek is cheating on her and tells her best friend Allison, who is the one sleeping with Derek. Corey and Pedro have been evicted, Pedro tries getting the rent in by stealing a car. Jessica has Problems with her boyfriend Aaron, after he accuses her of cheating when he contracted an STD. Everyone gets together at a party, which is later busted by the police, but not before Corey's sister is accidentally killed.

Q: *What do Heather and Allison do for a living?* A: They work with the ambulance/rescue squad.

Q: *Why is Aaron so upset with Jessica?*

A: He contracted an STD, and accuses her of cheating and giving it to him.

Q: *What is Allison hiding from Heather?*
A: Allison is sleeping with Heather's boyfriend Derek.

Q: *What does Jimmy do in Car's house?*
A: He comes around to pick up some drugs, but steals them when no one opens the door and he enters the open apartment.

Q: *What happens to Corey's sister?*
A: She goes to the party in the woods, where she is killed in a bathtub by a ricochet shot.

For the "Living in Oblivion", the following questions were asked: Q: *Who are the main characters of the movie?*
A: Nicole/Ellen (Catherine Keener), Nick (Steve Buscemi), Wolf (Dermot Mulroney), Palomino (James Le Gros), Wanda (Danielle von Zerneck), Cora (Rica Martens)

Q: *Please give a short overview of what you think the movie is about.*
A: An aspiring director and his crew are shooting an independent movie. Through dream sequences, we learn what Nick (the director) and Nicole (the lead actress) are affraid of, what they think can go wrong, and how they think the movie could fail. All kinds of problems plague the production, and a good deal of things go wrong until Cora (Nick's mother) saves the day by unknowingly playing the part of a "dwarf" who refuses to act.

Q: *What other movie has Nicole previously appeared in?*
A: She was in "that Richard Gere movie", where she was in a shower scene.

Q: *How many dream sequences are there?*
A: There are a total of three dream sequences, two actual dreams, and one "dream" that they film as part of the movie.

Q: *How does Nicole find out what Nick and Palomio think of her?*
A: She overhears them talking in another room during a break, via the boom operator's microphone.

Q: *Why is Wolf wearing an eyepatch?*
A: *Because he was struck by Wanda's cloths when she threw her shirt at him in the morning.*

Q: *How is Nick's movie saved in the end?*

A: Cora (unknowingly) plays the dwarf's part perfectly.

For the "Mumford", the following questions were asked: Q: *Who are the main characters of the movie?*
A: Mumford (Loren Dean), Lily (Alfre Woodard), Sofie (Hope Davis), Skip (Jason Lee); (also accepted Althea, Follet, Nessa, Lionel, Debanko, Sheeler)

Q: *Please give a short overview of what you think the movie is about.*
A: Mumford, a former investigator with the IRS who left because of drug problems, settles in a town named Mumford, pretending to be psychologist. He falls in love with Sofie, who is suffering from chronic fatigue syndrome, and plays match-maker for his neighbour Lily and local billionaire Skip. He also turns out to be quite a skilled "psychologist", simply through listening and giving good advice. He is being suspected by two actual doctors in the town, and eventually he is found out as he appears on a news program.

Q: *What is Mumford's secret, and who does he confide it to?*
A: He is not a psychologist at all, and only pretends and came to Mumford while running away from his past life. He tells Skip about it.

Q: *What is the relationship between Mumford and Lily?*
A: They are neighbours, and Mumford is a regular in her Cafe.

Q: *What did Mumford do for a living before he became a psychologist?*
A: He was an investigator with the IRS.

Q: *How is Mumford found out?*
A: His case is featured on the missing people TV show.

Q: *Who does Mumford fall in love with, and what is she suffering from?*
A: He falls in love with Sofie, who is suffering from chronic fatigue syndrome.

For the "A Nightmare on Elm Street 3: Dream Warriors", the following questions were asked: Q: *Who are the main characters of the movie?*
A: Nancy (Heather Langenkamp), Neil (Craig Wasson), Kristen (Patricia Arquette), Freddy (Robert Englund), Kincaid (Ken Sagoes), Joey (Rodney Eastman), Taryn (Jennifer Rubin) Phillip (Bradley Gregg), Will (Ira Heiden), Jennifer (Penelope Sudrow)

Q: *Please give a short overview of what you think the movie is about.*
A: A group of teenagers in a mental institute are being haunted in their dreams by

Freddy Kruger, a monstrous man who kills people in their sleep. Kirsten, who has recently been admitted, has the ability to pull others into her dreams, which the group eventually uses to fight off Freddy. Nancy, an intern at the hospital and the first person to actually fight Freddy and survive, and her boss help them once they find out what is really going on. Freddy is eventually defeated, but not before killing a number of the group members, including Nancy.

Q: *Who is Kirsten, what is her special gift?*
A: Kristen is a teenage girl who is in a mental institute as her parents believe her suicidal. Her special gift is that she can pull people into her dreams

Q: *Who is Phillip and what happens to him?*
A: Phillip is another patient in the mental institute, and a habitual sleepwalker. He gets thrown off the roof by Freddy, who makes it look like suicide.

Q: *What does Nancy reveal to the kids about Freddy?*
She tells them, that they are the children of the people who "killed" Freddy years ago (in the original movie).

Q: *How are Freddy's remains discovered and what happens to them?*
A: They are found and dug up in an auto salvage by Neil and Nancy's father. Neil eventually consecrates the remains.

Q: *How does Freddy deceive Nancy and what becomes of her?*
A: Freddy appears to her as her father to tell her that he is passing away (Freddy killed him). Nancy gets stabbed by Freddy.

For the "The Anniversary Party", the following questions were asked:
Q: *Who are the main characters of the movie?*
A: Joe (Alan Cumming), Sally (Jennifer Jason Leigh), Mac (John C. Reilly), Sophia (Phoebe Cates), Cal (Kevin Kline), Skye (Gwyneth Paltrow), Monica (Mina Badie), Ryan (Denis O'Hare)

Q: *Please give a short overview of what you think the movie is about.*
A: Joe and Sally are celebrating their 6th anniversary, being back together after a recent breakup. Some of the guest are their neighbours the Roses with whom they have a neighbourly war over their dog barking, Mac and Cal who are shooting a movie with Sally, Sophia who is Sally's best friend, and Skye, a rival actress of Sophia's who is starring in Joe's movie. They all take drugs and get high, during which time Mac almost drowns in the pool, the dog goes missing, and Joe and Sally fight. The party ends

completely when Joe's dad calls about his sister, who gave herself a drug overdose.

Q: *What is Sally's anniversary present for Joe?*
A: She bought a flat in London.

Q: *What do Sally and Joe do for a living?*
A: He is an author who is also directing the film adaptation of one of his books. She is a fading actress, who is shooting a movie with Mac and Cal.

Q: *What is the entertainment at the party?*
A: They play charades and take ecstasy.

Q: *What is Skye's present?*
A: She brings the drugs, "Dolphins".

Q: What happens to Mac?
A: He almost drowns in the pool while he is on drugs.

For the "We Own the Night", the following questions were asked:
Q: *Who are the main characters of the movie?*
A: Bobby (Joaquin Phoenix), Amanda (Eva Mendes), Joseph (Mark Wahlberg), Jumbo (Danny Hoch), Vadim (Alex Veadov), Marat (Moni Moshonov), Burt (Robert Duvall)

Q: *Please give a short overview of what you think the movie is about.*
A: Bobby and Joseph are two estranged brothers, the former owning a night club and living an according lifestyle, and the latter being in the police force like their father. Joseph tries to bust Vadim and his drug ring in Bobby's night club, but only a few arrests are made, and he falls victim to Vadim's retaliation, sending him into a coma. Bobby agrees to be an informant for the police, and eventually leads to them raiding a drug operation, arresting Vadim. Bobby has to go into police custody, but is betrayed and found. When his father is killed while transferring Vadim, Bobby decides to join the police force for good, and they take out the cartel.

Q: *What is Joseph's relationship to Bobby, what do Bobby and Joseph do for a living?*
A: Joseph and Bobby are brothers. Bobby is a nightclub manager, and Joseph is a police Captain.

Q: *Are the police successful at capturing Vadim when they raid the nightclub and how does he retaliate?*
A: They cannot arrest him during the raid, as he is not in possession of illegal goods. He retaliates for the raid by having Joseph shot and his cop car bombed.

Q: *What does Bobby do to avenge his father's death and what does Amanda do?*
A: Bobby decides to join the police force. Amanda is furious and leaves him, as he did not make plans with her and she counted on living the nightclub owner lifestyle.

Q: *Why was Joseph jealous of Bobby?*
A: He was jealous of his brother because Bobby was a free spirit and always did what he wanted, while Joseph always just followed their father.

Q: *Who betrayed Bobby and how does he find out?*
A: Louis betrayed him, telling people where Bobby lived in witness protection. Bobby finds out when Louis spills the beans accidentally after Bobby joint the police.

For the "While She Was Out", the following questions were asked:
Q: *Who are the main characters of the movie?*
A: Della (Kim Basinger), Chuckie (Lukas Haas), Huey (Jamie Starr), Vingh (Leonard Wu), Tomas (Luis Chavez)

Q: *Please give a short overview of what you think the movie is about.*
A: Della, a mother of two and married to an unbearable husband, drives off to the mall to meet with a friend and buy wrapping paper for Christmas. When she gets there, a Plymouth covers up two parking spots, even though the parking lot is very crowded, and she leaves the driver a note. After returning to the car, it turns out that the owners of the Plymouth are a violent gang, who shoot a rent-a-cop and then try to kill her. She flees to hide in the woods, where they chase her and she eventually kills them.

Q: *Why does Della leave for the Mall?*
A: She leaves the house to get away from her husband, and drives to the mall to get gift wrap.

Q: *What does Della do when she sees a Plymouth taking up two parking spots?*
A: She leaves a note on the car in anger.

Q: *What do the gang-members use to find Della?*
A: They use a construction site flash-light to find her in the night.

Q: *What does Della take with her from the car?*
A: She grabs her toolbox when she exits her car, fleeing from the gang.

Q: *What does Della do to Vingh and Tomas?*
A: Della kills Tomas with a wrench in a fight, flees from the others, and manages to sneak up on Vingh to kill him with a screwdriver.

Appendix C

Movie Summarisation Questionnaires

This appendix provides the questions given to participants of the first human evaluation study of Chapter 4.8.2. Participants saw the full movie, and were shown two summaries generated by the same modality (text OR video OR multi-modal), with different parameter settings.

Please answer the following questions regarding the full movie and the summaries you just saw.

1) In a few sentences, what do you think this movie is about?

Please answer the following questions, on a scale from 1 (not at all) to 5 (very much)

2) Overall, did you like the movie?

For Summary 1

3) Overall, does it provide a good summary of the movie?

4) Did the summary capture the important plot points of the movie?

5) Did the summary have a coherent, comprehensible structure?

6) Did the summary capture all parts (beginning, middle, end) of the movie?

For Summary 2

7) Overall, does it provide a good summary of the movie?

8) Did the summary capture the important plot points of the movie?

9) Did the summary have a coherent, comprehensible structure?

10) Did the summary capture all parts (beginning, middle, end) of the movie?

For the following, please choose either summary 1 or summary 2

11) Overall, which of the two summaries did you prefer?

Please provide answers for the following questions

12) For the summary you preferred, what was particularly good about it?

13) For the summary you preferred, was there anything that could have been better?

14) For the summary you dispreferred, what was particularly bad about it?

15) For the summary you dispreferred, was there anything that was still good about it?

16) (Optional) Do you have any other comments regarding the summaries, or this study in general?

Appendix D

Movie Overview Questionnaire

This appendix provides outline of the questionnaire used for the first Mechanical Turk based user study of Chapter 5.5.3, in which participants were asked to assess the quality of a single given movie overview.

Participants were issued the following instructions:

In this experiment, you will read a short movie summary (4 to 6 sentences) generated by a computer program.
The summary will discuss the following aspects of the movie:
1. MOOD: The overall tone of the movie, e.g. whether it is **humorous**, **bleak**, or **bittersweat**.
2. PLOT: The plot points the movie revolves around, such as **couple relations**, a **police investigation**, or **space travel**.
3. GENRES: The genres the movie can be classified as; for example **action**, **drama**, **comedy**.
4. ATTITUDES: The overall presentation of the movie, e.g, whether it is **fantastic** or **realistic**, or **fast** or **slow**.
5. PLACE: The country and/or location the movie takes place in, e.g. the **USA** or **Europe**, or a **hospital**.
6. FLAGS: The movie's content flags, for example whether it includes **violence** or **profane language**.
Your task is to judge whether the summary appropriately describes the movie. At the end of the task, we ask you to answer a question regarding the movie.
You must have seen the movie in order to complete this questionnaire.

As some movie titles can be ambiguous, we also provide a link to the movie's IMDB page. You may use this link to check whether the movie in question is one you have actually seen. However, please **do not attempt** to answer the questionnaire based on information provided by IMDB.
In cases where you are unsure or where a question does not apply, please answer "unsure".

MOVIE TITLE
Have you seen this movie? (yes/no)

Sentence 1.
Sentence 2.
Sentence 3.
Sentence 4.
Sentence 5.
Sentence 6.

1. Overall, does this summary **accurately describe the content** of the movie? (yes/no/unsure)

2. On a scale from 1 to 5, where 1 is "not at all" and 5 is "absolutely", **is this summary coherent**?
A **coherent** summary will be easy to understand, it will **cover most aspects** (e.g., mood, plot, genre) of the movie, and there will be no sentences that strike us as being in the wrong place. Summary sentences will be **ordered in a reasonable** way and will fit with their neighbouring sentences.

3. Does the **first** sentence apply to the movie's **mood**?
Reminder: Sentence 1 is [...]

4. On a scale from 1 to 5, where 1 is "not at all" and 5 is "absolutely", is the **first** sentence grammatical?

5. Does the **second** sentence apply to the movie's **plot**?
Reminder: Sentence 2 is [...]

6. On a scale from 1 to 5, where 1 is "not at all" and 5 is "absolutely", is the **second** sentence grammatical?

7. Does the **third** sentence apply to the movie's **genres**?
Reminder: Sentence 3 is [...]

8. On a scale from 1 to 5, where 1 is "not at all" and 5 is "absolutely", is the **third** sentence grammatical?

9. Does the **fourth** sentence apply to the movie's **attitude**?
Reminder: Sentence 4 is [...]

10. On a scale from 1 to 5, where 1 is "not at all" and 5 is "absolutely", is the **fourth** sentence grammatical?

11. Does the **fifth** sentence apply to the movie's **place**?
Reminder: Sentence 5 is [...]

12. On a scale from 1 to 5, where 1 is "not at all" and 5 is "absolutely", is the **fifth** sentence grammatical?

13. Does the **sixth** sentence apply to the movie's **content flags**?
Reminder: Sentence 5 is [...]

14. On a scale from 1 to 5, where 1 is "not at all" and 5 is "absolutely", is the **sixth** sentence grammatical?

15. If you can, please tell us what you think is good/bad about the provided summary (optional).

16. Please answer the following question for the movie:

Movie	Question	Example Answer
12 Monkeys	What is the big reveal at the end of the movie, and where does it happen?	It turns out that James Cole's "dream" was his childhood self witnessing the death of his time travelling adult self in the past. It happens at the airport.
Burn After Reading	What was Jason Osbourne's (John Malkovich) position within the CIA, and why did he quit eventually?	Jason was an analyst, who quit because he faced demotion for a drinking problem, and to have time to write his memoirs.
Gandhi	What is Gandhi's (Ben Kingsley) early key experience leading to his activism, and where does it happen?	In South Africa, he gets thrown off a train, and realises the latent racism in the country, especially against Indians.
Indiana Jones and the Kingdom of the Crystal Skull	Who is the mother of Indiana Jones' (Harrison Ford) son? Who are they pursued by?	Marion is the mother of his son, but Indiana did not know he even had one. They are pursued by the Soviet agents.
Midnight in Paris	What happens to Gil (Owen Wilson) in Paris, every night at Midnight?	He travels to different time periods, and hangs out with notable people.
Revolutionary Road	Where do Frank (Leonardo DiCaprio) and April (Kate Winslet) consider moving to, and why are they hesitating?	They consider moving to Paris. They reconsider, because Frank is offered a promotion, and April is pregnant.
Sideways	What are Miles' (Paul Giamatti) and Jack's (Thomas Haden Church) professions?	Miles is an unsuccessful screenwriter and an English teacher, and Jack is an actor.
So I Married an Axe Murderer	What is Harriet's (Nancy Travis) profession, and who is the actual Axe Murderer?	Harriet is a butcher. The real axe murderer is her sister.

The Bourne Identity	What is Jason Bourne's (Matt Damon) condition at the beginning and throughout the movie? Why was he shot by the CIA?	He is suffering from amnesia and does not know who he is. He was shot as a failed experiment on the loose.
The Rocky Horror Picture Show	Why do Brad (Barry Bostwick) and Janet (Susan Sarandon) enter the castle? What is Transylvania?	Their car broke down in the middle of nowhere, and the need a telephone. Transylvania is a far away galaxy.
The Ugly Truth	What is the titular "Ugly Truth", and how does Mike (Gerard Butler) help Abby (Katherine Heigl)?	The "Ugly Truth" is a local TV show. Mike helps Abby by giving her advice on how to get together with the man of her dreams.
Under Fire	Where does the movie take place, and what is happening there?	The movie takes place in Nicaragua, during the revolution that ended the Somoza regime.

Appendix E

Movie Overview Questionnaire

This appendix provides the outline of the questionnaire used for the second Mechanical Turk based user study of Chapter 5.5.3, in which participants were asked to comparatively rate six movie overviews that were generated by different systems.

Participants were issued the following instructions:

In this experiment, you will read 6 short movie summaries (4 to 6 sentences each) generated by different computer programs.
Your task is to judge whether the summaries **appropriately describe** the movie, as well as to **rank them according to which summary you think is best and worst**.
For some sentences, one or more systems may have failed to generate a sentence, instead producing **NONE**. This is fine, and just indicates that the system was not able to, for example, determine the location in which the movie takes place.
At the end of the task, we ask you to answer a question regarding the movie.
You must have seen the movie in order to complete this questionnaire. Please **do not attempt** the questionnaire **if you have not seen** the movie.
As some movie titles can be ambiguous, we also provide a link to the movie's IMDB page. You may use this link to check whether the movie in question is one you have actually seen. However, please **do not attempt** to answer the questionnaire based on information provided by IMDB.
In cases where you are unsure or where a question does not apply, please answer "unsure".

MOVIE TITLE

Have you seen this movie? (yes/no)

Please read the summaries below and then answer the following questions.

SUMMARY 1
SUMMARY 2
SUMMARY 3
SUMMARY 4
SUMMARY 5
SUMMARY 6

Please answer the following questions:

1. Overall, does SUMMARY 1 accurately describe the content of the movie? (yes/no/unsure)

2. Overall, does SUMMARY 2 accurately describe the content of the movie? (yes/no/unsure)

3. Overall, does SUMMARY 3 accurately describe the content of the movie? (yes/no/unsure)

4. Overall, does SUMMARY 4 accurately describe the content of the movie? (yes/no/unsure)

5. Overall, does SUMMARY 5 accurately describe the content of the movie? (yes/no/unsure)

6. Overall, does SUMMARY 6 accurately describe the content of the movie? (yes/no/unsure)

7. Please rank the above six summaries from BEST to WORST.

BEST	(Summary 1 / 2 / 3 / 4 / 5 / 6)
SECOND	(Summary 1 / 2 / 3 / 4 / 5 / 6)
THIRD	(Summary 1 / 2 / 3 / 4 / 5 / 6)
FOURTH	(Summary 1 / 2 / 3 / 4 / 5 / 6)
FIFTH	(Summary 1 / 2 / 3 / 4 / 5 / 6)
WORST	(Summary 1 / 2 / 3 / 4 / 5 / 6)

8. If you can, please tell us what you think is good/bad about the provided summaries (optional).

9. Please answer the following question for the movie

See Appendix D for movies, corresponding questions, and answers.

Bibliography

Agarwal, A., Balasubramanian, S., Zheng, J., and Dash, S. (2014a). Parsing Screenplays for Extracting Social Networks from Movies. In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature*, pages 50–58, Gothenburg, Sweden.

Agarwal, A., Dash, S., Balasubramanian, S., and Zheng, J. (2014b). Using Determinantal Point Processes for Clustering with Application to Automatically Generating and Drawing xkcd Movie Narrative Charts. In *Proceedings of the 2nd Academy of Science and Engineering International Conference on Big Data Science and Computing*, Stanford, California.

Agarwal, A., Zheng, J., Kamath, S., Balasubramanian, S., and Ann Dey, S. (2015). Key Female Characters in Film Have More to Talk About Besides Men: Automating the Bechdel Test. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 830–840, Denver, Colorado.

Angeli, G., Liang, P., and Klein, D. (2010). A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512. Association for Computational Linguistics.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*, San Diego, CA.

Bamman, D., O'Connor, B., and Smith, N. A. (2013). Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361, Sofia, Bulgaria.

Bamman, D., Underwood, T., and Smith, A. N. (2014). A Bayesian Mixed Effects

Model of Literary Character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 370–379, Baltimore, MD, USA.

Barzilay, R. and Lapata, M. (2005). Collective content selection for concept-to-text generation. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 331–338. Association for Computational Linguistics.

Barzilay, R. and Lapata, M. (2006). Aggregation via set partitioning for natural language generation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 359–366. Association for Computational Linguistics.

Barzilay, R. and McKeown, K. R. (2005). Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Belanger, D. and McCallum, A. (2016). Structured prediction energy networks. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 983–992, New York.

Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA.

Björkelund, A., Hafdell, L., and Nugues, P. (2009). Multilingual Semantic Role Labeling. In *Proceedings of the 13th Conference on Computational Natural Language Learning: Shared Task*, pages 43–48, Boulder, Colorado.

Borji, A., Cheng, M.-M., Jiang, H., and Li, J. (2015). Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722.

Bottou, L. (1991). Stochastic gradient learning in neural networks. In *Proceedings of Neuro-Nîmes 91*, Nimes, France. EC2.

Brox, T. and Malik, J. (2010). Object segmentation by long term analysis of point trajectories. *Computer Vision–ECCV 2010*, pages 282–295.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

Chen, L.-C., Schwing, A. G., Yuille, A. L., and Urtasun, R. (2015). Learning deep structured models. In *Proceedings of the 32nd International Comference on Machine Learning*, pages 1785–1794, Lille, France.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014a). On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014b). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chopra, S., Auli, M., and Rush, A. M. (2016). Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of NAACL: HLT*, pages 93–98, San Diego, CA.

Clarke, J. and Lapata, M. (2010). Discourse Constraints for Document Compression. *Computational Linguistics*, 36(3):411–441.

Cour, T., Jordan, C., Miltsakaki, E., and Taskar, B. (2008). Movie/script: Alignment and parsing of video and text transcription. *Computer Vision–ECCV 2008*, pages 158–171.

Dailianas, A., Allen, R. B., and England, P. (1995). Comparison of automatic video segmentation algorithms. In *Proc. SPIE*, volume 2615, pages 2–16.

Dalal, M. K. and Zaveri, M. A. (2011). Automatic Text Classification: A Technical Review. *International Journal of Computer Applications*, 28(2):37–40.

Danescu-Niculescu-Mizil, C. and Lee, L. (2011). Chameleons in Imagined Conversations: A New Approach to Understanding Coordination of Linguistic Style in Dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 76–87, Portland, Oregon.

DeJong, G. (1982). An overview of the FRUMP system. In Lehnert, W. G. and Ringle, M. H., editors, *Strategies for natural language processing*, pages 149–176. Lawrence Erlbaum Associates, Hillsdale, N. J.

Duboue, P. A. and McKeown, K. R. (2002). Content planner construction via evolutionary algorithms and a corpus-based fitness function. In *Proceedings of INLG 2002*, pages 89–96.

Elsner, M. (2012). Character-based kernels for novelistic plot structure. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 634–644, Avignon, France.

Elson, D. K., Dames, N., and McKeown, K. R. (2010). Extracting Social Networks from Literary Fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, Uppsala, Sweden.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Evangelopoulos, G., Zlatintsi, A., Potamianos, A., Maragos, P., Rapantzikos, K., Skoumas, G., and Avrithis, Y. (2013). Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention. *IEEE Transactions on Multimedia*, 15(7):1553–1568.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9 (Aug):1871–1974.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Godbole, S. and Sarawagi, S. (2004). Discriminative methods for multi-labeled classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30. Springer.

Goldberg, E., Driedger, N., and Kittredge, R. I. (1994). Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53.

Gorinski, P. J. and Lapata, M. (2015). Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, Denver, Colorado.

Hearst, M. A. (1997). Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1):33–64.

Hershey, J. R., Roux, J. L., and Weninger, F. (2014). Deep unfolding: Model-based inspiration of novel deep architectures. *CoRR*, abs/1409.2574.

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

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep structured output learning for unconstrained text recognition. *CoRR*, abs/1412.5903.

Jones, K. S. et al. (1999). Automatic summarizing: factors and directions. *Advances in automatic text summarization*, pages 1–12.

Ju, S. X., Black, M. J., Minneman, S., and Kimber, D. (1998). Summarization of videotaped presentations: automatic analysis of motion and gesture. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):686–696.

Kasturi, R., Strayer, S. H., Gargi, U., and Antani, S. (1996). An evaluation of color histogram based methods in video indexing. In *International workshop on image database and multi media search, Amsterdam, The Netherlands*, pages 75–82.

Kim, J.-S., Sim, J.-Y., and Kim, C.-S. (2014). Multiscale Saliency Detection Using Random Walk With Restart. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(2):198–210.

Koncel-Kedziorski, R., Konstas, I., Zettlemoyer, L., and Hajishirzi, H. (2016). A theme-rewriting approach for generating algebra word problems. *arXiv preprint arXiv:1610.06210*.

Konstas, I. and Lapata, M. (2012). Concept-to-text generation via discriminative reranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 369–378. Association for Computational Linguistics.

Kurata, G., Xiang, B., and Zhou, B. (2016). Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence. In *HLT-NAACL*.

Lebret, R., Grangier, D., and Auli, M. (2016). Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas.

Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., and Jurafsky, D. (2011). Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In *Proceedings of the 15th Conference on Computational Natural Language Learning: Shared Task*, pages 28–34, Portland, OR, USA.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Lienhart, R., Pfeiffer, S., and Effelsberg, W. (1997). Video abstracting. *Communications of the ACM*, 40(12):54–62.

Lin, C., Tsai, C., Kang, L., and Lin, W. (2013). Scene-Based Movie Summarization via Role-Community Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(11):1927–1940.

Lin, C.-Y. and Hovy, E. (2000). The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 495–501. Association for Computational Linguistics.

Lloret, E. and Palomar, M. (2009). A gradual combination of features for building automatic summarisation systems. In *International Conference on Text, Speech and Dialogue*, pages 16–23. Springer.

Lu, W. and Ng, H. T. (2011). A probabilistic forest-to-string model for language generation from typed lambda calculus expressions. In *Proceedings of the Conference on*

Empirical Methods in Natural Language Processing, pages 1611–1622. Association for Computational Linguistics.

Lu, Z. and Grauman, K. (2013). Story-Driven Summarization for Egocentric Video. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2714–2721, Portland, OR, USA.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. pages 674–679.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Ma, Y.-F., Hua, X.-S., Lu, L., and Zhang, H.-J. (2005). A generic framework of user attention model and its application in video summarization. *IEEE transactions on multimedia*, 7(5):907–919.

Ma, Y.-F., Lu, L., Zhang, H.-J., and Li, M. (2002). A user attention model for video summarization. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 533–542. ACM.

Mani, I., Klein, G., House, D., Hirschman, L., Firmin, T., and Sundheim, B. (2002). SUMMAC: A Text Summarization Evaluation. *Natural Language Engineering*, 8(1):43–68.

Mani, I. and Maybury, M. T. (1999). *Advances in automatic text summarization*, volume 293. MIT Press.

Mann, H. B. and Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Mathias, M., Benenson, R., Pedersoli, M., and Van Gool, L. (2014). Face detection without bells and whistles. In *ECCV*.

Mei, H., Bansal, M., and Walter, M. R. (2016). What to talk about and how? selective generation using lstms with coarse-to-fine alignment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 720–730, San Diego, California.

Michel, J.-B., Shen, Y. K., aviva Presser Aiden, Veres, A., Gray, M. K., Team, T. G. B., Pickett, J. P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M. A., and Aiden, E. L. (2010). Quantitative Analysis of Culture Using Millions of Digitized Books. *Science*, 331(6014):176–182.

Monaco, J. (1982). *How to Read a Film: The Art, Technology, Language, History and Theory of Film and Media*. OUP, New York, NY, USA.

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

Mori, T. (2002). Information gain ratio as term weight: the case of summarization of ir results. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Morris, A., Kasper, G., and Adams, D. (1992). The Effects and Limitations of Automated Text Condensing on Reading Comprehension Performance. *Information Systems Research*, 3(1):17–35.

Mosteller, F. and Wallace, D. (1964). *Inference and Dispituted Authorship: The Federalists*. Addison-Wesley, Boston, MA, USA.

Nalisnick, T. E. and Baird, S. H. (2013). Character-to-Character Sentiment Analysis in Shakespeare's Plays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 479–483, Sofia, Bulgaria.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

Nelken, R. and Shieber, S. (2006). Towards Robust Context-Sensitive Sentence Alignment for Monolingual Corpora. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 161–168, Trento, Italy.

Neto, J. L., Santos, A. D., Kaestner, C. A., Alexandre, N., Santos, D., et al. (2000). Document clustering and text summarization.

Nielsen, F. (2010). Afinn-96. *Department of Informatics and Mathematical Modelling, Technical University of Denmark*.

Nielsen, F. A. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big Things Come in Small Packages*, pages 93–98, Heraklion, Crete.

Noreen, E. (1989). *Computer Intensive Methods for Testing Hypotheses: An Introduction*. Wiley, Hoboken, New Jesey.

Otsuji, K., Tonomura, Y., and Ohba, Y. (1991). Video browsing using brightness data. In *Visual Communications,'91, Boston, MA*, pages 980–989. International Society for Optics and Photonics.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number SIDL-WP-1999-0120.

Papazoglou, A. and Ferrari, V. (2013). Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1777–1784.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Park, S.-B., Kim, H.-N., Kim, H., and Jo, G.-S. (2010). Exploiting script-subtitles alignment to scene boundary dectection in movie. In *Multimedia (ISM), 2010 IEEE International Symposium on*, pages 49–56. IEEE.

Parkhi, O. M., Vedaldi, A., Zisserman, A., et al. (2015). Deep face recognition. In *BMVC*, volume 1, page 6.

Pfeiffer, S., Lienhart, R., Fischer, S., and Effelsberg, W. (1996). Abstracting digital movies automatically. *Journal of Visual Communication and Image Representation*, 7(4):345–353.

Rasheed, Z., Sheikh, Y., and Shah, M. (2005). On the Use of Computable Features for Film Classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1):52–64.

Read, J., Pfahringer, B., and Holmes, G. (2008). Multi-label classification using ensembles of pruned sets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 995–1000. IEEE.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359.

Reed, T., editor (2004). *Digital Image Sequence Processing*. Taylor & Francis.

Reiter, E. and Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Reiter, E., Mellish, C., and Levine, J. (1995). Automatic generation of technical documentation. *Applied Artificial Intelligence an International Journal*, 9(3):259–287.

Ren, R., Misra, H., and Jose, J. M. (2010). Semantic based adaptive movie summarisation. In *MMM*, pages 389–399. Springer.

Sang, J. and Xu, C. (2010a). Character-based Movie Summarization. In *Proceedings of the International Conference on Multimedia*, pages 855–858, Firenze, Italy.

Sang, J. and Xu, C. (2010b). Character-based movie summarization. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 855–858. ACM.

Sankar, P., Jawahar, C. V., and Zisserman, A. (2009). Subtitle-free movie to script alignment. In *British Machine Vision Conference*.

Sap, M., Prasettio, M. C., Holtzman, A., Rashkin, H., and Choi, Y. (2017). Connotation frames of power and agency in modern films. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2319–2324.

Schwing, A. G. and Urtasun, R. (2015). Fully connected deep structured networks. *CoRR*, abs/1503.02351.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *Association for Computing Machinery: Computing Surveys*, 34(1):1–47.

Smeaton, A. F., Lehane, B., O'Connor, N. E., Brady, C., and Craig, G. (2006). Automatically selecting shots for action movie trailers. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 231–238. ACM.

Smoliar, S. W. and Zhang, H. (1994). Content based video indexing and retrieval. *IEEE multimedia*, 1(2):62–72.

Snyder, B. (2005). *Save the Cat!: The Last Book on Screenwriting You'll Ever Need*. M. Wiese Productions, Studio City, CA.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Stoyanov, V., Ropson, A., and Eisner, J. (2011). Empirical risk minimization of graphical model parameters given approximate inference, decoding, and model structure. In *AISTATS*, pages 725–733.

Sundberg, P., Brox, T., Maire, M., Arbeláez, P., and Malik, J. (2011). Occlusion boundary detection and figure/ground assignment from optical flow. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2233–2240. IEEE.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Tang, D., Qin, B., and Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

Teodosio, L. and Bender, W. (1993). Salient video stills: Content and context preserved. In *Proceedings of the first ACM international conference on Multimedia*, pages 39–46. ACM.

Tomasi, C. and Kanade, T. (1991). Detection and tracking of point features. Technical report, International Journal of Computer Vision.

Tong, H., Faloutsos, C., and Pan, J.-Y. (2006). Fast Random Walk with Restart and Its Applications. In *Proceedings of the Sixth International Conference on Data Mining*, pages 613–622, Hong Kong.

Tsoumakas, G. and Katakis, I. (2006). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3).

Tsoumakas, G. and Vlahavas, I. (2007). Random k-labelsets: An ensemble method for multilabel classification. In *European Conference on Machine Learning*, pages 406–417. Springer.

Vedaldi, A. and Lenc, K. (2015). Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2):137–154.

Wang, F. and Ngo, C.-W. (2007). Rushes video summarization by object and event understanding. In *Proceedings of the international workshop on TRECVID video summarization*, pages 25–29. ACM.

Wen, T.-H., Gasic, M., Mrkšić, N., Su, P.-H., Vandyke, D., and Young, S. (2015). Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal.

Wen, T.-H., Gašić, M., Mrkšić, N., Rojas-Barahona, L. M., Su, P.-H., Vandyke, D., and Young, S. (2016). Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 120–129, San Diego, California.

Weng, C., Chu, W.-T., and ling Wu, J. (2009). Rolenet: Movie Analysis from the perspective of Social Networks. *IEEE Transactions on Multimedia*, 11(2):256–271.

Wolf, W. (1996). Key frame selection by motion analysis. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 2, pages 1228–1231. IEEE.

Wu, B., Zhang, Y., Hu, B.-G., and Ji, Q. (2013). Constrained clustering and its application to face clustering in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3507–3514.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Ye, P. and Baldwin, T. (2008). Towards Automatic Animated Storyboarding. In *Proceedings of the 23rd Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, pages 578–583, Chicago, Illinois.

Zhang, M.-L. and Zhou, Z.-H. (2005). A k-nearest neighbor based algorithm for multi-label classification. In *Granular Computing, 2005 IEEE International Conference on*, volume 2, pages 718–721. IEEE.

Zhang, M.-L. and Zhou, Z.-H. (2006). Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans. on Knowl. and Data Eng.*, 18(10):1338–1351.

Zhang, X. and Lapata, M. (2014). Chinese poetry generation with recurrent neural networks. In *EMNLP*, pages 670–680.

Zhang, X. and Lapata, M. (2017). Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Copenhagen, Denmark.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. (2015). Conditional random fields as recurrent neural networks.

In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537.

Zhou, H., Hermans, T., Karandikar, A. V., and Rehg, J. M. (2010). Movie genre classification via scene categorization. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 747–750, New York, NY.

Zhou, L. and Hovy, E. (2004). Template-filtered Headline Summarization. In *Proceedings of the Association for Computational Linguistics workshop: Text Summarization Branches Out*, pages 56–60.