# Movie Script Summarization as Graph-based Scene Extraction

**Philip John Gorinski** and **Mirella Lapata**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
P.J.Gorinski@sms.ed.ac.uk, mlap@inf.ed.ac.uk

## Abstract

In this paper we study the task of movie script summarization, which we argue could enhance script browsing, give readers a rough idea of the script's plotline, and speed up reading time. We formalize the process of generating a shorter version of a screenplay as the task of finding an optimal chain of scenes. We develop a graph-based model that selects a chain by jointly optimizing 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.

```
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 1: Excerpt from "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 lines describe what the camera sees (e.g., `We can't get a good glimpse of his face, but his body...`).

## 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 analyze 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 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

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

*Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pages 1066–1076,
Denver, Colorado, May 31 – June 5, 2015. ©2015 Association for Computational Linguistics

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 paper we explore whether current NLP technology can be used to address some of these tasks. Specifically, we focus on script summarization, which we conceptualize 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 formalize 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 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 optimizing logical progression, diversity, and importance.

Our contributions in this work are three-fold: we introduce a novel summarization task, on a new text genre, and formalize scene selection as the problem of finding a chain that represents a film's story; we propose several novel methods for analyzing 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.

## 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 digitized 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) analyze 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) analyzes 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.

Although we are not aware of any previous approaches to summarize screenplays, the field of computer vision is rife with attempts to summarize video (see Reed 2004 for an overview). Most techniques are based on visual information and rely on low-level cues such as motion, color, 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. Characters are typically identified using face recognition techniques and scene boundaries are presumed unknown and are automatically detected. A notable exception are Sang and Xu (2010) who generate video summaries for movies, while taking into account character interaction features which they estimate from the corresponding screenplay.

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,

|  | # Movies | AvgLines | AvgScenes | AvgChars |
|---|---|---|---|---|
| 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 |

Figure 2: ScriptBase corpus statistics. Movies can have multiple genres, thus numbers do not add up to 1,276.

social interactions, and interests. Lu and Grauman (2013) present a summarization model which 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). 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 ScriptBase: A Movie Script Corpus

We compiled ScriptBase, a collection of 1,276 movie scripts, by automatically crawling web-sites which host or link entire movie scripts (e.g., imsdb.com). The retrieved scripts were then cross-matched against Wikipedia[2] and IMDB[3] and paired with corresponding user-written summaries, plot sections, loglines and taglines (taglines are short snippets used by marketing departments to promote a movie). We also collected meta-information regarding the movie's genre, its actors, the production year, etc. ScriptBase contains movies comprising 23 genres; each movie is on average accompanied by 3 user summaries, 3 loglines, and 3 taglines. The corpus spans years 1909–2013. Some corpus statistics are shown in Figure 2.

The scripts were further post-processed with the Stanford CoreNLP pipeline (Manning et al., 2014) to perform tagging, 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). Our summarization experiments focused on comedies and thrillers. We randomly selected 30 movies for training/development and 65 movies for testing.

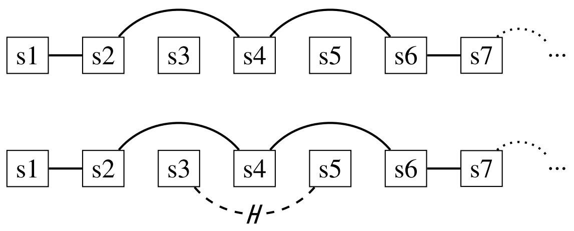

Figure 3: Example of consecutive chain (top). Squares represent scenes in a screenplay. The bottom chain would not be allowed, since the connection between $s3$ and $s5$ makes it non-consecutive.

## 4 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 1). 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 a set $S = \{s_1, s_2, \ldots, s_n\}$ of scenes, and a set $C = \{c_1, \ldots, c_m\}$ of characters. We are interested in finding a list $S' = \{s_i, \ldots s_k\}$ of *ordered, consecutive* scenes subject to a compression rate $m$ (see the example in Figure 3). 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 story; and (c) scenes which highlight the story's *progression* from beginning to end. We therefore find the chain $S'$ maximizing the objective function $Q(S')$ which is the weighted sum of three terms: the story progression $P$, scene diversity $D$, and scene importance $I$:

$$S^* = arg\max_{S' \subset S} Q(S') \tag{1}$$

$$Q(S') = \lambda_P P(S') + \lambda_D D(S') + \lambda_I I(S') \tag{2}$$

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

[2] http://en.wikipedia.org

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

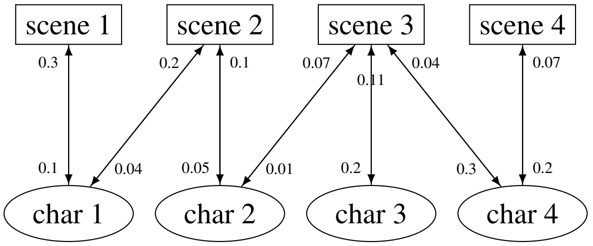

Figure 4: Example of a bipartite graph, connecting a movie's scenes with participating characters.

**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 operationalize 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 $s_{i+1}$:

$$P(S') = \sum_{i=0}^{|S'|-1} \sum_{c \in C_i} INF(s_i, s_{i+1}|c) \tag{3}$$

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 a bipartite graph is shown in Figure 4. 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 weights between a character and a scene. We are interested in how important a character is generally in the movie, and specifically in a particular scene. For $w_{c,s}$, we consider the probability of a character being important, i.e., of them belonging to the set of main characters:

$$w_{c,s} = P(c \in main(M)), \, \forall (c, s, w_{c,s}) \in E \tag{4}$$

where $P(c \in main(M))$ is some probability score associated with $c$ being a main character in script $M$. For $w_{s,c}$, we take the number of interactions a character is involved in relative to the total number of interactions in a specific scene as indicative of the character's importance in that scene. Interactions refer to conversational interactions as well as relations between characters (e.g., who does what to whom):

$$w_{s,c} = \frac{\sum\limits_{c' \in C_s} inter(c, c')}{\sum\limits_{c_1, c_2 \in C_s} inter(c_1, c_2)}, \, \forall (s, c, w_{s,c}) \in E \tag{5}$$

We defer discussion of how we model probability $P(c \in Main(M))$ and obtain interaction counts to Section 5. Weights $w_{s,c}$ and $w_{c,s}$ are normalized:

$$w_{s,c} = \frac{w_{s,c}}{\sum_{(s,c',w'_{s,c})} w'_{s,c}}, \, \forall (s, c, w_{s,c}) \in E \tag{6}$$

$$w_{c,s} = \frac{w_{c,s}}{\sum_{(c,s',w'_{c,s})} w'_{c,s}}, \, \forall (c, s, w_{c,s}) \in E \tag{7}$$

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

$$T(i, j) = \begin{cases} w_{i,j} \text{ if } (v_i, v_j, w_{i,j} \in E^B) \\ 0 \text{ otherwise} \end{cases} \tag{8}$$

We measure the *influence* 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:

$$r_k = (1 - \varepsilon) T r_k + \varepsilon e_k \tag{9}$$

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 $\varepsilon$ 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[4] by solving:

$$r_k = \varepsilon(I-(1-\varepsilon)T)^{-1}e_k \tag{10}$$

The $l$th 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:

$$INF(s_i,s_{i+1}|c) = r_{s_i}[s_{i+1}] - r_{s_i}^c[s_{i+1}] \tag{11}$$

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) 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. The diversity of chain $S'$ is the sum of the diversities of its successive scenes:

$$D(S') = \sum_{i=1}^{|S'|-1} d(s_i,s_{i+1}) \tag{12}$$

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:

$$d(s_i,s_{i+1}) = \frac{d_{char}(s_i,s_{i+1}) + d_{sen}(s_i,s_{i+1})}{2} \tag{13}$$

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

$$d_{char}(s_i,s_{i+1}) = 1 - \frac{|C_{s_i} \cap C_{s_{i+1}}|}{|C_{s_i} \cup C_{s_{i+1}}|} \tag{14}$$

$d_{char}$ 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:

$$d_{sen}(s_i,s_{i+1}) = 1 - \frac{k \cdot dif(s_i,s_{i+1})}{k - k \cdot dif(s_i,s_{i+1}) + 1} \tag{15}$$

$$dif(s_i,s_{i+1}) = \frac{1}{1+|sen(s_i) - sen(s_{i+1})|} \tag{16}$$

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

$$sen(s) = \sum_{c,c' \in C_s} sen(inter(c,c')) \tag{17}$$

We explain how interactions and their sentiment are computed in Section 5. Again, $d_{sen}$ is larger if two scenes have a less similar sentiment. $dif(s_i,s_{i+1})$ becomes 1 if the sentiments are identical, and increasingly smaller for more dissimilar sentiments. The sigmoid-like function in Equation (15) scales $d_{sen}$ within range $[0,1]$ to take smaller values for larger sentiment differences (factor $k$ adjusts the curve's smoothness).

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

$$I(S') = \sum_{i=1}^{|S'|} imp(s_i) \tag{18}$$

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

$$imp(s_i) = \frac{\sum_{c:\, c \in C_{s_i} \wedge c \in main(M)} 1}{\sum_{c:\, c \in C_{s_i}} 1} \tag{19}$$

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.[5] $I(s_i)$ is 0 if a scene does not contain any main characters, and 1 if it contains only main characters (see Section 5 for how $main(M)$ is inferred).

**Optimal Chain Selection** We use Linear Programming to efficiently find a good chain. The objective is to maximize Equation (2), i.e., the sum of the terms for progress, diversity and importance,

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

[5] 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.

subject to their weights λ. We add a constraint corresponding to the compression rate, i.e., the number of scenes to be selected and enforce their linear order by disallowing non-consecutive combinations. We use GLPK[6] to solve the linear problem.

## 5 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 $w_{s,c}$ and $w_{c,s}$ in the bipartite graph.

**Interactions** The notion of 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 (5)), and in defining diversity (see Equations (15)–(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*. Speaker attribution comes for free in our case, as speakers are clearly marked in the text (see Figure 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 relations from the output of a semantic role labeler. Relations are denoted by verbs whose ARG0 and ARG1 roles are character names. We extract relations from the dialogue but also from scene descriptions. For example, in Figure 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).

**Sentiment** We labeled 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 (17)), and even the sentiment between characters (e.g., who likes or dislikes whom in the movie in general).

**Main Characters** The progress term in our summarization objective crucially relies on characters and their importance (see the weight $w_{c,s}$ in Equation (4)). Previous work (Weng et al., 2009; Lin et al., 2013) 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:

$$\begin{aligned} G = \{C, E\}, : C = \{c_1, \dots c_n\}, \\ E = \{(c_i, c_j, w) | c_i, c_j \in C, \, w \in \mathbb{N}_{>0}\} \end{aligned}$$

where vertices correspond to movie characters[7], and edges denote character-to-character interactions. Figure 5 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 weights are not normalized, 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

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

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

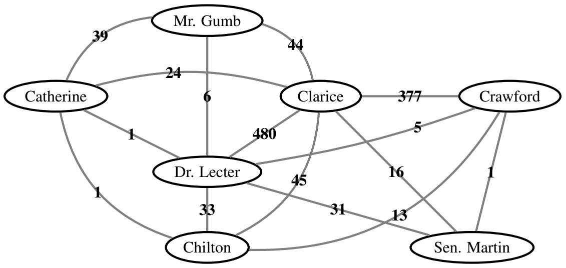

Figure 5: Social network for "The Silence of the Lambs"; edge weights correspond to absolute number of interactions between nodes.

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 assuming that the characters listed under Wikipedia's *Cast* section (or an equivalent section, e.g., *Characters*) are the main characters in the movie.

Examples of the features we used for the 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 these features, the MLP achieves an F1 of 79.0% on the test set. It outperforms other classification methods 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 (4), lead characters also appear in Equation (19), which determines scene importance. We assume a character $c \in main(M)$ if it is predicted by the MLP with a probability $\geq 0.5$.

## 6 Experimental Setup

**Gold Standard Chains** The development and tuning of the chain extraction model presented in Section 4 necessitates access to a gold standard of key scene chains representing the movie's most important content. Our experiments concentrated on a sample of 95 movies (comedies and thrillers) from the ScriptBase corpus (Section 3). 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 in a movie. Using the alignment algorithm presented in Nelken and Shieber (2006), we align script sentences to Wikipedia plot sentences and assume that scenes with at least one alignment are part of the *gold chain* of scenes. We obtain many-to-many alignments using features such as lemma overlap and word stem similarity. When evaluated on four movies[8] (from the training set) whose content was manually aligned to Wikipedia plots, the aligner achieved a precision of .53 at a recall rate of .82 at deciding whether a scene should be aligned. Scenes are ranked according to the number of alignments they contain. When creating gold chains at different compression rates, we start with the best-ranked scenes and then successively add lower ranked ones until we reach the desired compression rate.

**System Comparison** In our experiments we compared our scene extraction model (SceneSum) against three baselines. The first baseline was based on the *minimum* 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 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 optimized 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

[8] "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 attending high 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 1: Questions for the movie "One Eight Seven".

to $\varepsilon = 0.5$, and the sigmoid scaling factor in our diversity term to $k = -1.2$.

**Evaluation** 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 summarizing 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 1 for an example). In addition, for every movie we asked subjects to name the main characters, and summarize its plot (in no more than four sentences). Using Amazon Mechanical Turk (AMT)[9], we elicited answers for eight scripts (four comedies and thrillers) in 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 screenplays 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.

| | 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 |
| SceneSum | 0.23 | 0.37 | 0.50 | 0.60 | 0.68 |
| Random | 0.10 | 0.20 | 0.30 | 0.40 | 0.50 |

Table 2: Model performance on automatically generated gold standard (test set) at different compression rates.

## 7 Results

Table 2 shows the performance of SceneSum, our scene extraction model, and the three comparison systems (MaxOv, MinOv, Random) on the automatic gold standard at five compression rates. As can be seen, MaxOv performs best in terms of F1, followed by SceneSum. We believe this is an artifact due to the way the gold standard was created. Scenes with large numbers of main characters are more likely to figure in Wikipedia 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.

We further analyzed the scenes selected by SceneSum and the comparison systems with respect to their position in the script. Table 3 shows the av-

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

|  | Beginning | Middle | End |
|---|---|---|---|
| MaxOv | 33.95 | 34.89 | 31.16 |
| MinOv | 34.30 | 33.91 | 31.80 |
| SceneSum | 35.30 | 33.54 | 31.16 |
| Random | 34.30 | 33.91 | 31.80 |

Table 3: Average percentage of scenes taken from the beginning, middle and ends of movies, on automatic gold standard test set.

erage percentage of scenes selected from the beginning, middle, and end of the movie (based on an equal division of the number of scenes in the screenplay). As can be seen, the number of selected scenes tends to be evenly distributed across the entire movie. SceneSum has a slight bias towards the beginning of the movie which is probably natural, since leading characters appear early on, as well as important scenes introducing essential story elements (e.g., setting, points of view).

The results of our human evaluation study are summarized in Table 4. We observe that SceneSum summaries are overall more informative compared to those created by the baselines. In other words, AMT participants are able to answer more questions regarding the story of the movie when reading SceneSum summaries. 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 and 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 unfavorable scenes. Table 4 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.

| 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 4: Percentage of questions answered correctly.

## 8 Conclusions

In this paper we have developed a graph-based model for script summarization. We formalized the process of generating a shorter version of a screenplay as the task of finding an optimal chain of scenes, which are diverse, important, and exhibit logical progression. A large-scale evaluation based on a question-answering task revealed that our method produces more informative summaries compared to several baselines. In the future, we plan to explore model performance in a wider range of movie genres as well as its applicability to other NLP tasks (e.g., book summarization or event extraction). We would also like to automatically determine the compression rate which should presumably vary according to the movie's length and content. Finally, our long-term goal is to be able to generate loglines as well as movie plot summaries.

**Acknowledgments** We would like to thank Rik Sarkar, Jon Oberlander and Annie Louis for their valuable feedback. Special thanks to Bharat Ambati, Lea Frermann, and Daniel Renshaw for their help with system evaluation.

## References

Bamman, David, Brendan O'Connor, and Noah A. Smith. 2013. Learning Latent Personas of

Film Characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria, pages 352–361.

Bamman, David, Ted Underwood, and A. Noah Smith. 2014. A Bayesian Mixed Effects Model of Literary Character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, MD, USA, pages 370–379.

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

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

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

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

Kim, Jun-Seong, Jae-Young Sim, and Chang-Su Kim. 2014. Multiscale Saliency Detection Using Random Walk With Restart. *IEEE Transactions on Circuits and Systems for Video Technology* 24(2):198–210.

Lee, Heeyoung, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 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*. Portland, OR, USA, pages 28–34.

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

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

Mani, Inderjeet, Gary Klein, David House, Lynette Hirschman, Therese Firmin, and Beth Sundheim. 2002. SUMMAC: A Text Summarization Evaluation. *Natural Language Engineering* 8(1):43–68.

Manning, Christopher, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 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.

Michel, Jean-Baptiste, Yuan Kui Shen, aviva Presser Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aiden. 2010. Quantitative Analysis of Culture Using Millions of Digitized Books. *Science* 331(6014):176–182.

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

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

Mosteller, Frederick and David Wallace. 1964. *Inference and Dispituted Authorship: The Federalists*. Addison-Wesley, Boston, MA, USA.

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

Nelken, Rani and Stuart Shieber. 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*. Trento, Italy, pages 161–168.

Nielsen, Finn Arup. 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*. Heraklion, Crete, pages 93–98.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number SIDL-WP-1999-0120.

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

Reed, Todd, editor. 2004. *Digital Image Sequence Processing*. Taylor & Francis.

Sang, Jitao and Changsheng Xu. 2010. Character-based Movie Summarization. In *Proceedings of the International Conference on Multimedia*. Firenze, Italy, pages 855–858.

Tong, Hanghang, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast Random Walk with Restart and Its Applications. In *Proceedings of the Sixth International Conference on Data Mining*. Hong Kong, pages 613–622.

Weng, Chung-yi, Wei-Ta Chu, and Ja ling Wu. 2009. Rolenet: Movie Analysis from the perspective of Social Networks. *IEEE Transactions on Multimedia* 11(2):256–271.