Automated Screenplay Annotation for Extracting Storytelling Knowledge

David R. Winer[1] and **R. Michael Young**[2]
[1,2]School of Computing
[2]Entertainment Arts and Engineering Program
University of Utah
Salt Lake City, UT 84112
drwiner@cs.utah.edu, young@eae.utah.edu

Abstract

Narrative screenplays follow a standardized format for their parts (e.g., stage direction, dialogue, etc.) including short descriptions for what, where, when, and how to film the events in the story (shot headings). We created a grammar based on the syntax of shot headings to extract this and other discourse elements for automatic screenplay annotation. We test our annotator on over a thousand raw screenplays from the IMSDb screenplay corpus and make the output available for narrative intelligence research.

Expert knowledge for storytelling is difficult to formalize and hand-code, causing an *authorial bottleneck problem* (Valls-Vargas, Zhu, and Ontanon 2016; Riedl and Sugandh 2008). Despite a long research history of encoding communicative knowledge as actions (Cohen and Perrault 1979; Young and Moore 1994; Jhala and Young 2010), it isn't widely practiced in storytelling and resources are limited. Recent work with information extraction has shown progress for informing narrative models from unannotated narrative text (Chambers and Jurafsky 2008; Goyal, Riloff, and Daumé III 2010; Valls-Vargas, Zhu, and Ontanon 2016) and from crowdsourced examples (Li et al. 2013). However, these models are still impoverished in the kinds of features they use related to communication and scene-structure (e.g., sentiment analysis (Li et al. 2014; Reagan et al. 2016)).

Screenplays (film scripts) are unusual for narrative text because they contain more structured discourse information (Jhala 2008) than other narrative texts such as news stories (Chambers and Jurafsky 2008) or fables (Goyal, Riloff, and Daumé III 2010; Valls-Vargas, Zhu, and Ontanon 2016) which have received more attention. Screenplays follow a standardized format for their parts (e.g., stage direction, dialogue, etc.) including short descriptions for what, where, when, and how to film the events in the story (shot headings). According to *The Hollywood Standard* (Riley 2009), an authoritative guide to screenplay writing, these shot headings follow a rigid syntax which we've formalized and implemented as part of a screenplay parser. There are many screenplays available online potentially provide a wealth of information to inform a model of storytelling for different scenarios and genres.

In this paper, we introduce the elements of screenplays, reveal our strategy for parsing these elements, and discuss the progress of this project including results on parsing raw screenplays from the Internet Movie Script Database (IMSDb)[1] whose annotations are made available for subsequent research (Winer 2017). Then, we present our approach for extracting storytelling knowledge using the parser's output in the context of our high-level vision for narrative understanding.

Related Work

The idea of leveraging the standardized format of screenplays is not new. Jhala (2008) proposes automating movie script annotations, citing the conventions and standard structure of movie scripts as motivation for extracting the wealth of information they contain. An early approach by Vassiliou (2006) describes grammar-based templates for extracting categories of events in screenplays such as scene changes, changes to location, and non-verbal communication. Scene changes are detected by first locating the words *day* or *night*, then search left for *INT* or *EXT* (and label everything between *INT*/*EXT* and *day*/*night* as a location). More recently, Agarwal and colleagues (2014) experiment with machine learning approaches for classifying each line in a screenplay as either a scene boundary, stage direction, character name, dialogue, or transition. They show machine learning approaches perform better than a rule-based approach, citing difficulty in the rule-based approach for detecting shot headings when there is no *INT*/*EXT* present or when the heading is not capitalized.

We found two main areas of research that have published on extracting character dialogue from screenplays.

- In *computational social science*, the evolution of character relationships in screenplays are mapped to a computational model of social networks using features like scene co-occurrence and sentiment analysis (Qu et al. 2015). Tsoneva and colleagues (2007) use these networks to help automatically summarize movies. They segment screenplays by shot headings and extract speaker - dialogue pairs for each segment, but do not use information embedded in the shot headings.

[1]www.imsdb.com

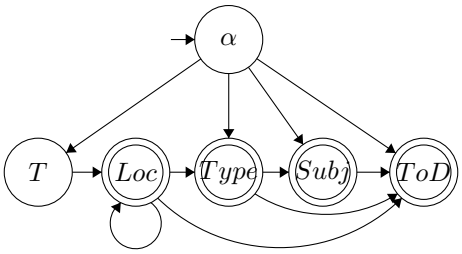

Figure 1: Simplified shot heading automata starting at α

- In *conversational dialogue systems*, character dialogue is used to train models for dialogue agents. Walker and colleagues (2012) annotate dialogue from the IMSDb corpus to characterize and learn conversational personas for story characters. Li and colleagues (2016) use character dialogue and subtitles in television series transcripts to train speaker consistency in open-domain conversation.

Also relevant is the Movie Script Markup Language (Van Rijsselbergen et al. 2009) which was developed to computerize screenplays to facilitate interactive collaboration among the various parties involved with producing film; it is a language for creating screenplays which handles the story logic and other internal pragmatics of shooting film.

Screenplay Elements

There are 4 basic format elements in screenplays (Riley 2009):

1. *Shot Headings* - begins each new scene or shot, and may give general information about a scene's location, type of shot, subject of shot, or time of day
2. *Direction* - what is being seen or heard within the shot or scene, may also include words or phrases *in caps*
3. *Dialogue* - name of character and actual words which are spoken, including parenthetical character direction related to the dialogue
4. *Transitions* - may appear at end of scene, indicates how one scene links to the next

Shot Headings

According to *The Hollywood Standard* (Riley 2009), an authoritative reference for screenplay formatting, shot headings can include up to five pieces of information:

1. INT., EXT., or INT./EXT (interior or exterior)
2. Location (increasingly more specific locations separated by hyphens)
3. Shot type (e.g., TRACKING, MED. SHOT, ANGLE ON)
4. Subject of the shot (a person, a place, an object, etc.)
5. Time of day (e.g., a date, NOON, DEAD OF NIGHT, etc.)

We cover how these elements are combined (see Figure 1 for a simplified automata representing how these elements are combined). For simplicity, the first two pieces of information will henceforth be called the *setting*.

A *master shot heading* starts with a setting, and then may finish with a time of day, such as:

```
INT. CENTRAL PARK - DAY
```

Shot types may be followed by a hyphen or a preposition indicating the way the shot is capturing the subject, such as "CLOSE - X" vs. "CLOSE ON X". Also, each piece of information may include "modifiers" (words or dates which appear in parentheses).

```
EXT. WHITE HOUSE - SOUTH LAWN - CLOSE
ON CNN CORRESPONDENT - SUNSET (MARCH
15, 1999)
```

The use of INDY'S RUN in the following example from *Indiana Jones and the Raiders of the Lost Ark* (Kasdan and Lucas 1981) is unconventional because actions do not typically appear in shot headings, and only locations would appear before a shot type, so INDY'S RUN should be interpreted as a sub-location of THE JUNGLE.

```
EXT. THE JUNGLE - INDY'S RUN - CLOSE
ANGLE - DAY
```

Shot headings can also start with a shot type:

```
WIDE SHOT - RACETRACK AND EMPTY STANDS
```

sometimes on its own:

```
TRACKING SHOT
```

or the header can start with a time of day on its own:

```
DEAD OF NIGHT
```

or include just a subject of the shot:

```
RUDOLF
```

Master shot headings are sometimes interpreted as the beginning of a scene, whereas other shot headings (without a setting) are sub-scenes.

Grammar We created a grammar which models the syntax of a shot heading according to our best interpretation of *The Hollywood Standard* (Riley 2009):

$$
\begin{aligned}
&\alpha \rightarrow Scene \mid Shot \mid SUB \mid (ST + Opt\text{–}ToD) \\
&Scene \rightarrow Setting + Opt\text{–}Shot \\
&Setting \rightarrow T + Opt\text{–}H + place + Loc \\
&T \rightarrow \text{`INT.'} \mid \text{`EXT.'} \mid \text{`INT./EXT.'} \\
&Shot \rightarrow ST + Opt\text{–}P + SUB \\
&Loc \rightarrow \text{`–'} + (Loc \mid place) + Opt\text{–}M \\
&SUB \rightarrow Subj + Opt\text{–}ToD \\
&Subj \rightarrow (place \mid person \mid obj) + Opt\text{–}M \\
&ToD \rightarrow time + Opt\text{–}M \\
&ST \rightarrow type + Opt\text{–}M \\
&Mod \rightarrow \text{`('} + WORDS\_digits + \text{`)'}
\end{aligned}
$$

$Opt{-}Shot \rightarrow (\text{`}-\text{'} + (Shot \mid ToD)) \mid \{\}$
$Opt{-}M \rightarrow Mod \mid \{\}$
$Opt{-}P \rightarrow Opt{-}H \mid prep$
$Opt{-}ToD \rightarrow (\text{`}-\text{'} + ToD) \mid \{\}$
$Opt{-}H \rightarrow \text{`}-\text{'} \mid \{\}$

$prep \in$ {"on, "with", "to", "towards", "from", "in", "under", "over", "above", "around", "into"}
$type \in$ set with enumerated shot types
$place, person, obj, time \rightarrow$ names or expressions

In-Line Caps

Some words in stage direction are capitalized which we refer to as *in-line caps*; in a subset of cases, this notation indicates that the word or words in caps are an entity played by an actor/actress appearing on screen for the first time in the film and indicates they will have some dialogue. Other reasons for using in-line caps are to describe sound effects or off-screen sounds (including the thing that makes the sound) or for denoting camera actions that appear in stage direction. As Riley (2009) points out, a new shot heading isn't necessary for camera movements such as panning or zooming.

```
INT. SUBMARINE - GALLEY - NIGHT

Nason and his guys fight the fire. They
are CHOKING on smoke. PAN TO Ensign
Menendez, leading in a fresh contingent
of men to join the fight. One of them
is TITO.
```

In this example[2], the shot includes a pan movement of the camera to Ensign Menendez, we hear the sound of choking, and Tito is a character whose actor is shown on screen for the first time.

Transitions

Transitions indicate how a shot or scene changes to the next (Riley 2009). The only transition that occurs on the left-hand side of the page is "FADE IN"; otherwise, transitions are distinctively indented to the right-hand side. Transitions have different functions: "FADE OUT", "FADE TO BLACK", and "CUT TO BLACK" usually indicate the end of the film. Other types include cuts and dissolves (also wipes, but these seem to be less functional).

A cut is an instantaneous shift and is the most common shot transition; however, they are typically only explicitly written into a screenplay when they are being used to emphasize or contrast. *The Hollywood Standard* provides several examples such as:

```
Jenny smells the rose.

                                HARD CUT TO:

SCREAMING LOCOMOTIVE
```

[2]Example modified from *The Hollywood Standard* (Riley 2009)

and

```
The fat lady opens her mouth to sing.

                                MATCH CUT TO:

LITTLE SALLY SALTER
```

A dissolve is a gradual transition from one image to another which may imply a passage of time or entering into a character's imagination (Riley 2009).

Dialogue

Dialogue has distinctive indenting. This example is from *Indiana Jones and the Raiders of the Lost Ark* (Kasdan and Lucas 1981):

```
As the mine car is about to disappear
into the tunnel -

                INDY

          (to Marion)

     Get down!
```

Page Formatting

Filmmakers tacitly agree to use the convention that one script page translates to one minute of finished film, on average (Riley 2009). As a result, the spacing and formatting of a page may reflect the pace of the action and the number of minutes into the film the action is occurring. Screenwriters use heuristics and strategies about the pacing of action in films (e.g., Story Map (Calvisi 2011)). Screenplay pages are 57 lines.

Parsing Screenplays

We have developed a tool for this project called `ScreenPy` (Winer 2017) to parse screenplay elements which uses the actively maintained PyParsing module (McGuire 2006), a recursive descent parser.

Definition 1 (Shot Heading) *A parsed shot heading is a tuple of the form* $\langle T, Loc, ST, Subj, ToD, \tau \rangle$ *where* $T \in \{$*'INT.'*, *'EXT.'*, *'INT./EXT.'*, $\emptyset\}$, Loc *is a list of increasingly specific locations (or empty),* ST *is a shot type from from an enumerated list of types (or empty),* $Subj$ *is a word or phrase (or blank),* ToD *is a word or phrase for the time of day (or blank), and* τ *is the starting and stopping index of the heading in the screenplay when read as a text file. Shot headings must have at least one non-empty item. A shot heading is a* ***master*** *heading just when* T *is nonempty.*

The key insight to implementing the grammar for shot headings is that the accuracy relies on classifying ST and ToD. A Loc is extracted just when it's preceded by INT./EXT., and a Loc cannot be followed immediately by a $Subj$ (which would make it difficult to detect the Loc -

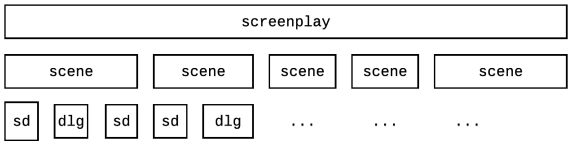

Figure 2: A schematic of the hierarchical structure of screenplay segments; 'sd' = stage direction, 'dlg' = dialogue.

$Subj$ boundary). `ScreenPy` is loaded with an exhaustively enumerated set of camera shot keywords such as "WIDE", "TRACKING", and "TILT" to detect ST words and phrases which do not contain the word "SHOT". We also look for prepositional keywords (e.g., 'ON', 'TO', 'WITH', etc.) as a ST - $Subj$ boundaries, as in "CLOSE ON CNN CORRESPONDENT".

`ScreenPy` use a multi-tier approach to detect if a shot heading element is a ToD (e.g., "3 AM" or "MIDMORNING") rather than a $Subj$.

1. First, we use the heading context: if the word or phrase is not the last element of the shot heading, then it cannot be a ToD, and if it follows two hyphens from ST, then it must be a ToD. This leaves instances when the shot heading is one element or the element in question follows a shot type. If it follows Loc and is not a ST, then it also must be a ToD because $Subj$ cannot follow Loc.
2. Second, we use Python's datetime module's *is_date* method as inclusive criteria.
3. Third, the word or phrase may have a keyword or keyphrase from enumerated list (e.g., 'sunrise', 'sunset', 'dusk', 'birthday', 'annual', 'christmas'), then it is included.
4. If it is neither a datetime nor on the enumerated list, then we use the *sense2vec* module (Trask, Michalak, and Liu 2015) to rate the similarity of the word or phrase to the word "time", and if the score is above an arbitrary threshold "0.55" (which is pretty similar), then we also label it as ToD. If it fails these criteria and is not a ST, then it is a $Subj$.

Temporal information extraction is an active area of research (e.g., `SUTime` (Chang and Manning 2012)) and future implementations of `ScreenPy` should use an approach or off-the-shelf system for better accuracy and to extract the meaning of the ToD and fit it into a timeline.

`ScreenPy` decomposes the screenplay into a set of primitive-level **segments** (see Figure 2). It extracts speaker - dialogue pairs based on the center-indented title of the speaker and extra indent of the dialogue and treat these as a special dialogue-typed segments. Center-indented titles which are not followed by dialogue are actually titles and are ignored.

Definition 2 (Segment) *A **segment** is a tuple of the form $\langle H, E, C\rangle$ where either H is a shot heading and E is stage direction or H is a speaker and E is dialogue, and C is the set of capitalized words or phrases in E. If s is a segment of the form $\langle H, E, C\rangle$, let $head(s) = H$, $text(s) = E$, and $caps(s) = C$.*

If the shot heading is a **master** heading, then it is a scene which is hierarchically above non-master segments.

Definition 3 (Master Segment) *A **master segment** is a 2-tuple $\langle m_{head}, S\rangle$ where m_{head} is a master shot heading and $S = s_0, ..., s_n$ indicating that m_{head} is followed by n shot segments s_1 through s_n, $head(s_0) = m_{head}$, and if m_{head} is followed by stage direction E, then $text(s_0) = E$ and otherwise $text(s_0) = \emptyset$. If m is a master segment of the form $\langle h, S\rangle$, a **parent link** written $m \lhd s$ indicates that $s \in S$.*

`ScreenPy` extracts transitions using their right-side indent and treat these as labeled edges between segments. A **transition** between segments s_i, s_{i+1} with label t (e.g., "CUT TO") is written $s_i \underset{t}{\rightarrow} s_{i+1}$.

Definition 4 (Screenplay) *A **screenplay** is a tuple of the form $\langle S, \Delta, P\rangle$ where S is an ordered list of segments (including master segments), Δ is a set of transitions between segments in S, and P is a set of parent links between segments in S. If m is a master segment in parent link $m \lhd s$, then $m \prec s$ in S.*

The following segment from the screenplay of *Indiana Jones and the Raiders of the Lost Ark* (Kasdan and Lucas 1981) consists of a shot heading following by stage direction.

```
AT THE VINED LANDING

Indy sails through sideways and rolls
to a stop at the bottom of the steps.
His whip is grasped in his hand. As he
raises himself, he hears, from above
the giant spikes of the Chamber of
Light CLANG! and an abrupt, sickening
rendition of SATIPO'S LAST SCREAM. Indy
runs up the steps. The rumbling sound
grows louder.
```

The resulting JSON object for this segment (the output of our parser) appears in Figure 3. The "start" and "stop" indices refer to positions in the screenplay text when read as a string.

Data Collection

We have made available a corpus of screenplays where each scene is parsed into its constituent parts in the form of a JSON object as described in the previous section. We collected 1068 raw screenplays from IMSDb. A first round of JSON objects have been produced using `ScreenPy` (Winer 2017).

About 8% of the screenplays could not be parsed because they had little or no formatting such as the indentations that would help distinguish shot headings from speakers, which is needed to distinguish dialogue from stage direction. Of the

Table 1: Analysis on parser's output on IMSDb screenplays by genre (screenplay can have multiple genres). Includes avg numbers (per screenplay in genre) for master headings (msegs), segments (segs), segments with shot heading (headings), dialogue segments (speakers), segs with shot type (has shot), segs with subject (has subj), and segs with time of day (has tod).

GENRE	FILMS	msegs	segs	headings	speakers	has shot	has subj	has tod
Action	272	145.32	1240.44	621.07	538.46	26.28	207.86	89.86
Adventure	143	135.96	1201.85	574.01	546.62	19.32	181.02	80.64
Animation	24	83.42	1190.79	466.08	674.71	18.96	116.88	47.92
Biography	3	144.33	1352.67	543.00	756.00	8.33	57.67	123.00
Comedy	310	114.37	1370.38	581.92	720.32	20.25	174.33	78.83
Crime	193	135.07	1352.31	656.12	620.73	18.33	185.67	93.49
Drama	541	123.10	1328.10	591.28	666.60	20.36	161.21	85.77
Family	22	110.45	1385.86	740.59	561.36	36.95	208.23	59.64
Fantasy	90	120.06	1213.94	627.40	526.83	29.60	227.74	73.33
Film-Noir	4	49.50	1535.00	721.75	786.50	139.25	201.50	13.00
History	3	107.33	1302.00	594.00	600.67	1.67	175.33	95.00
Horror	134	124.38	1149.94	631.57	450.96	32.89	238.72	79.32
Music	5	110.40	1179.20	503.40	660.80	7.00	53.60	96.40
Musical	14	96.21	1388.93	712.64	606.29	27.00	208.43	71.57
Mystery	97	137.91	1294.98	608.71	621.03	14.27	168.92	90.93
Romance	177	116.37	1367.10	596.67	710.13	25.08	176.86	84.27
Sci-Fi	140	142.34	1161.19	606.56	471.83	23.66	202.91	79.14
Short	3	39.00	269.33	138.33	118.00	8.33	32.67	25.67
Sport	2	229.00	1380.00	881.00	499.00	3.00	647.00	71.00
Thriller	352	135.31	1265.74	626.78	574.90	25.98	216.25	86.01
War	25	105.76	1272.40	616.08	577.88	26.36	263.68	85.88
Western	10	145.70	1367.10	706.00	602.10	76.10	223.50	103.30
sum	2564	2651.29	27569.26	13344.96	12891.72	608.97	4329.98	1713.95
avg	116.55	120.51	1253.15	606.59	585.99	27.68	196.82	77.91

92%, occasional errors are found. The reasons for these errors are still being reviewed to debug the parser. Sometimes these are due to differences in the formatting or digits in areas which refer to revisions to the draft (and are therefore rectifiable). We plan to account for these variations as best as possible and iteratively re-release the corpus as `ScreenPy` is improved.

We summarize some basic findings for our parser's output on the basis of the segments extracted in Table 1. These stats will be used to help find errors in our segmentation.

Knowledge Extraction

We propose to extract storytelling patterns from screenplays to address the authorial bottleneck problem referenced in the introduction. The overall goal is to learn hierarchical cinematic narrative discourse patterns for narrative generation (Jhala and Young 2010; Young et al. 2013) where an instantiated pattern would represent a generated segment whose *subplan* consists of actions taken by characters and whose preconditions and effects include conditions associated with features of the segment. These patterns would then be used for *automated cinematic narrative generation*. We will focus specifically on segments containing stage direction and leave dialogue for future work. The task of learning plan-based operators is broken into stages: 1] *lexical action recognition*, and 2] *schema induction*. In this paper, we describe the first stage and provide definitions relevant to the second.


```
"segment": {
    "indent": 15, "start": 21308, "stop": 21360,
    "heading": "AT THE VINED LANDING",
    "text":  {"stop": 21763, "start": 21360,
        "type": "direction"
        "raw_text": "Indy sails through
sideways and rolls to a stop at the
bottom of the steps. His whip is grasped
in his hand. As he raises himself, he
hears, from above the giant spikes of
the Chamber of Light CLANG! and an
abrupt, sickening rendition of SATIPO'S
LAST SCREAM. Indy runs up the steps. The
rumbling sound grows louder."}
    "in_lines": ["CLANG", "SATIPO'S LAST SCREAM"],
}
```


Figure 3: JSON object resulting from the parse of a screenplay fragment from *Indiana Jones and the Raiders of the Lost Ark.*

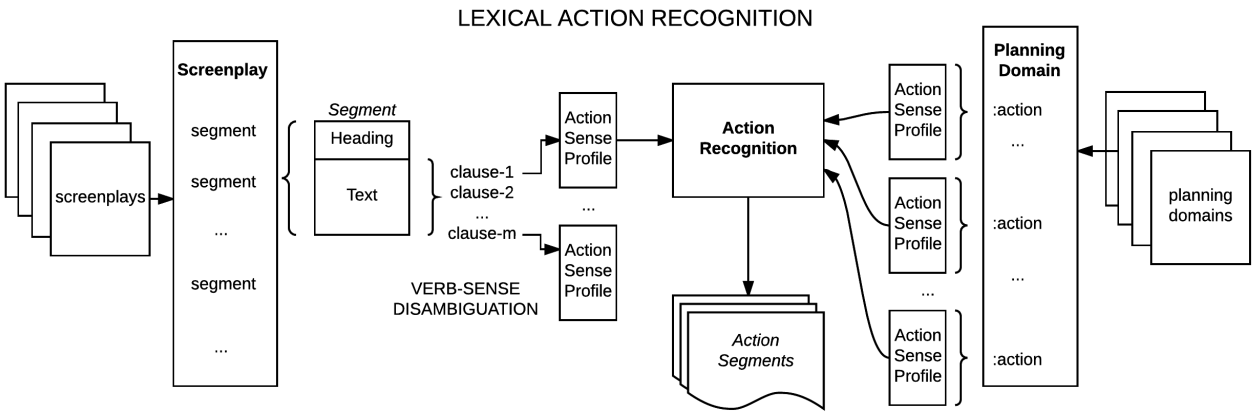

Figure 4: Schematic of the lexical action recognition process

Lexical Action Recognition

After collecting segments as in Figure 4, we propose to map clauses in stage direction to STRIPS-style (Fikes and Nilsson 1972) action schemata for a domain of interest provided as input. The planning domain provided as input may be tailored for a specific genre (e.g., Western shootouts, dragon slaying, etc.) or represent a generic set of action types that can occur in a wide variety of contexts. Action schemata in a planning domain are manually annotated with a set of lexical constraints.

Definition 5 (Action) *An **action schema** is a tuple of the form $\langle t, V, a, P, E\rangle$ where t is an action name, V is an ordered list of typed variables, $a \in V \cup \emptyset$ is an agent which performs the action, P is a set of function-free literal preconditions, and E is a set of function-free literal effects. If s is an action schema of the form $\langle t, V, a, P, E\rangle$, let $eff(s) = E$ and $pre(s) = P$.*

It would not be sufficient to label action schemata with specific verbs because verbs have a variety of meanings. In computational linguistics, *word-sense disambiguation* is the task of identifying which sense of a word is used in a sentence when a word has multiple meanings. This task tends to be more difficult with verbs than nouns because verbs have more senses on average than nouns and may be part of a multiword phrase (Del Corro, Gemulla, and Weikum 2014). Verb sense disambiguation (VSD) is aided by syntactic pruning (a verb sense may be limited to a number of syntactic patterns), and semantic pruning (a verb sense is limited to a number of semantic argument types) (Del Corro, Gemulla, and Weikum 2014).

We leverage two widely used lexical databases: FrameNet (Baker, Fillmore, and Lowe 1998) and WordNet (Fellbaum 2010). The FrameNet database uses *frames* which are schematic representations of types of situations such as an action's operation and relationships between arguments. It includes 1,200 frames, but not all are associated with verbs or verb phrases (and we are only interested in those which are). We use an off-the-shelf frame-semantic parser `Semafor` (Das and Smith 2011) to identify verb and argument frames from an input sentence.

However, frames are insufficient for the task because they do not commit to a particular instance of a situation. For example, the *Cause_change_of_position_on_scale* frame, which means that an agent or cause affects of the position of an item on some scale from one value to another, does not indicate if the movement of that value is to increase or decrease. For this reason, we also use the WordNet database which groups words into *synsets*, categories representing synonyms which can be shared among verbs. We use the off-the-shelf clause parser `ClausIE` (Del Corro and Gemulla 2013) to identify clauses in a sentence and to label the type of clause from set {SV, SVA, SVC, SVO, SVOO, SVOA, SVOC} where S is subject, V is verb, O is object, C is complement, and A is adverbial. We use the clause type to prune the set of possibly synsets for a verb instance as done in other work (Del Corro, Gemulla, and Weikum 2014).

We use WordNet synsets and FrameNet frames to manually characterize action schemata.

Definition 6 (Action Sense Profile) *If t is an action or clause, then t's action sense profile is a tuple of the from $\langle S_t, F_t\rangle$ where S_t is a set of synsets associated with the sense of verb uses which can represent t and F_t is a set of frames associated with the intended category for the operation of t.*

Given a text segment x and planning domain with action schemata O each annotated with an action sense profile, we assign schemata in O to verbs in x. Figure 4 shows a schematic of the lexical action recognition process (*lexical* as opposed to visual action recognition such as in video (Liu, Luo, and Shah 2009)). The criteria for an assignment should be minimally that if v is a verb instance and o is an action schema, then assign o to v just when $|S_v \cap S_o| > 0 \wedge |F_v \cap F_o| > 0$.

An example output of the VSD process given the segment in Figure 3 is shown in Figure 5. We have not yet annotated action schemata with sense profiles needed to com-

```
"segment": {
    "indent": 15, "start": 21308, "stop": 21360,
    "heading": "AT THE VINED LANDING",
    "clauses": [
        {"clause": "Indy sails through sideways"
        "verb": "sail"},
        "synsets": [<"sweep.v.02">],
        "frame": Change_direction},
        {"clause": Indy rolls to a stop at the
                bottom of the steps",
        "verb": "rolls",
        "synsets": [... ],
        "frame": ... }
    ...]
}
```


Figure 5: Abbreviated JSON object from Figure 3 after verb sense disambiguation.

plete the action recognition process. After action recognition, if clause c is paired with an action schema type a, then $|F_c \cap F_a| > 0$ and these frame(s) are used to define bindings between arguments of c and parameters of a for substitution. The type of c (e.g., SVO) is also used to inform binding decisions.

The instantiated action schema is an **action instance**. An action instance is *partial* just when at least one argument is not substituted.

Definition 7 (Action Segment) *Given a segment $s = \langle H, E, C \rangle$, an **action segment** representing s is a tuple of the form $\langle H, A, H_a, C_a \rangle$ where A is an ordered list of action instances extracted from E, H_a is the set of parameters in actions in A which are substituted by elements of H, and C_a is the set of parameters in actions in A which are substituted by capitalized words or phrases in C.*

Schema Induction

The second stage for learning hierarchical patterns representing screenplay segments is *schema induction*. Our methodology will be inspired by similar work such as learning narrative scripts from commonly co-occurring verb instances in news stories (Chambers and Jurafsky 2008) and learning from crowdsourced stories about the same event (Li et al. 2013). The details for this process are still in development, so this section only provides definitions relevant to defining features that would play a role in our approach.

Action segments are mapped to a vector representation. Features, the positions on the vector, are defined relative to an action segment and its local context. A feature is created for each action schemata in the input planning domain and for potential causal links between actions.

Definition 8 (Potential Causal Link) *Two partial action instances a_i, a_j are in a **potential causal link**, denoted $a_i \xrightarrow{p} a_j$, just when $a_i \prec a_j$ in the text, $\exists e, p'$ where $e \in eff(a_i)$, $p' \in pre(a_j)$ and p is the most-general-unifier of e and p', and $\neg\exists a'$ s.t. $a_i \prec a' \prec a_j$ and $\neg p \in eff(a')$.*

Features are created for each action parameter representing the entities of focus in a segment.

Definition 9 (Focus) *The **focus** of an an action segment s of the form $\langle H, A, H_a, C_a \rangle$ are $H_a \cup C_a$. Two sequences s, s' where $s = \langle H', A', H'_a, C'_a \rangle$ **share focus** just when $|H'_a \cap H_a| > 0$ or $|C'_a \cap C_a| > 0$ or $subj = subj'$ where $subj \in H$ and $subj' \in H'$.*

Camera shot types are binned into categories (e.g., *close*, *wide*, *medium*, *tracking*, *dolly*, *pan*, etc.) and a feature is created for each type.

Features are created for each type of transition and for the hierarchical structure of the scene (e.g., two segments which are recipients of a parent link from the same master segment are in the same scene).

Locations in the shot headings are hierarchically structured. Two locations are *siblings* when they are both sub-locations of the same general location. A location is a *parent* of another if one is a sub-location of another.

A segment heading which has no ToD takes on the ToD from the previous segment. Two action segments are at the *same time* just when they have the same ToD. Future work would benefit from binning changes to ToD into categories reflecting meaningful differences in time.

Conclusion

We presented the structural elements of screenplays, introduced a parsing strategy and tool to automate annotation, discussed our current status for data collection, and proposed future work for extracting storytelling knowledge. The major contributions described in this paper are a) the identification of the structure of information in shot headings, b) the description of a means of extracting that information, and c) a technical agenda for extracting storytelling knowledge from screenplay segments. The approach implements a parser whose grammar follows the authoritative guide for writing screenplays. The resulting corpus and the parser are accessible for download. We hope to expand the corpus and to improve the tool to account for our errors.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1654651, for which the authors are thankful.

References

Agarwal, A.; Balasubramanian, S.; Zheng, J.; and Dash, S. 2014. Parsing screenplays for extracting social networks from movies. In *Proceedings of the European Association for Computational Linguistics*, 50–58.

Baker, C. F.; Fillmore, C. J.; and Lowe, J. B. 1998. The berkeley framenet project. In *36th ACL and 17th COLING*, 86–90. Proceedings of the Association for Computational Linguistics.

Calvisi, D. 2011. *Story Maps: How to Write a Great Screenplay*. Smashwords Edition.

Chambers, N., and Jurafsky, D. 2008. Unsupervised learning of narrative event chains. In *Proceedings of the Association for Computational Linguistics*, volume 94305, 789–797.

Chang, A. X., and Manning, C. D. 2012. Sutime: A library for recognizing and normalizing time expressions. In *Proceedings of The International Conference on Language Resources and Evaluation*, volume 2012, 3735–3740.

Cohen, P. R., and Perrault, C. R. 1979. Elements of a plan-based theory of speech acts. *Cognitive Science* 3(3):177–212.

Das, D., and Smith, N. A. 2011. Semi-supervised frame-semantic parsing for unknown predicates. In *49th ACL: Human Language Technologies*, 1435–1444. Proceedings of the Association for Computational Linguistics.

Del Corro, L., and Gemulla, R. 2013. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, 355–366. ACM.

Del Corro, L.; Gemulla, R.; and Weikum, G. 2014. Werdy: Recognition and disambiguation of verbs and verb phrases with syntactic and semantic pruning. In *Proceedings of the Association for Computational Linguistics*.

Fellbaum, C. 2010. Wordnet. In *Theory and applications of ontology: computer applications*. Springer. 231–243.

Fikes, R. E., and Nilsson, N. J. 1972. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2(3):189–208.

Goyal, A.; Riloff, E.; and Daumé III, H. 2010. Automatically producing plot unit representations for narrative text. In *EMNLP*, 77–86. Proceedings of the Association for Computational Linguistics.

Jhala, A., and Young, R. M. 2010. Cinematic visual discourse: Representation, generation, and evaluation. *IEEE Transactions on Computational Intelligence and AI in Games* 2(2):69–81.

Jhala, A. 2008. Exploiting structure and conventions of movie scripts for information retrieval and text mining. *Interactive Storytelling* 210–213.

Kasdan, L., and Lucas, G. 1981. Raiders of the lost ark. http://www.imsdb.com/scripts/Indiana-Jones-and-the-Raiders-of-the-Lost-Ark.html.

Li, B.; Lee-Urban, S.; Johnston, G.; and Riedl, M. 2013. Story Generation with Crowdsourced Plot Graphs. In *AAAI*.

Li, B.; Thakkar, M.; Wang, Y.; and Riedl, M. O. 2014. Storytelling with adjustable narrator styles and sentiments. In *Proceedings of the International Conference on Interactive Digital Storytelling*, 1–12. Springer.

Li, J.; Galley, M.; Brockett, C.; Spithourakis, G. P.; Gao, J.; and Dolan, B. 2016. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.

Liu, J.; Luo, J.; and Shah, M. 2009. Recognizing realistic actions from videos "in the wild". In *Computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on*, 1996–2003. IEEE.

McGuire, P. 2006. Introduction to pyparsing: An object-oriented easy-to-use toolkit for building recursive descent parsers. *PyCon*.

Qu, W.; Zhang, Y.; Wang, D.; Feng, S.; and Yu, G. 2015. Semantic movie summarization based on string of ie-rolenets. *Computational Visual Media* 1(2):129–141.

Reagan, A. J.; Mitchell, L.; Kiley, D.; Danforth, C. M.; and Dodds, P. S. 2016. The emotional arcs of stories are dominated by six basic shapes. *EPJ Data Science* 5(1):31.

Riedl, M., and Sugandh, N. 2008. Story planning with vignettes: Toward overcoming the content production bottleneck. *Interactive Storytelling* 168–179.

Riley, C. 2009. *The Hollywood standard: the complete and authoritative guide to script format and style*. Michael Wiese Productions.

Trask, A.; Michalak, P.; and Liu, J. 2015. Sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings. *arXiv preprint arXiv:1511.06388*.

Tsoneva, T.; Barbieri, M.; and Weda, H. 2007. Automated summarization of narrative video on a semantic level. In *ICSC*, 169–176. IEEE.

Valls-Vargas, J.; Zhu, J.; and Ontanon, S. 2016. Error analysis in an automated narrative information extraction pipeline. *IEEE Transactions on Computational Intelligence and AI in Games*.

Van Rijsselbergen, D.; Van De Keer, B.; Verwaest, M.; Mannens, E.; and Van de Walle, R. 2009. Movie script markup language. In *9th ACM symposium on Document engineering*, 161–170. ACM.

Vassiliou, A. 2006. *Analysing Film Content: A Text-Based Approach*. Ph.D. Dissertation, University of Surrey.

Walker, M. A.; Lin, G. I.; and Sawyer, J. 2012. An annotated corpus of film dialogue for learning and characterizing character style. In *Proceedings of The International Conference on Language Resources and Evaluation*, 1373–1378.

Winer, D. R. 2017. Screenpy: an automated screenplay annotation tool. https://www.github.com/drwiner/screenpy.

Young, R. M., and Moore, J. D. 1994. DPOCL: A principled approach to discourse planning. In *INLG*, 13–20. Proceedings of the Association for Computational Linguistics.

Young, R. M.; Ware, S.; Cassell, B.; and Robertson, J. 2013. Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives. *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative* 37(1-2):41–64.