Automatically Producing Plot Unit Representations for Narrative Text

Amit Goyal
Dept. of Computer Science
University of Maryland
College Park, MD 20742
amit@umiacs.umd.edu

Ellen Riloff
School of Computing
University of Utah
Salt Lake City, UT 84112
riloff@cs.utah.edu

Hal Daumé III
Dept. of Computer Science
University of Maryland
College Park, MD 20742
hal@umiacs.umd.edu

Abstract

In the 1980s, plot units were proposed as a conceptual knowledge structure for representing and summarizing narrative stories. Our research explores whether current NLP technology can be used to automatically produce plot unit representations for narrative text. We create a system called AESOP that exploits a variety of existing resources to identify affect states and applies "projection rules" to map the affect states onto the characters in a story. We also use corpus-based techniques to generate a new type of affect knowledge base: verbs that impart positive or negative states onto their patients (e.g., being eaten is an undesirable state, but being fed is a desirable state). We harvest these "patient polarity verbs" from a Web corpus using two techniques: co-occurrence with Evil/Kind Agent patterns, and bootstrapping over conjunctions of verbs. We evaluate the plot unit representations produced by our system on a small collection of Aesop's fables.

1 Introduction

In the 1980s, plot units (Lehnert, 1981) were proposed as a knowledge structure for representing narrative stories and generating summaries. Plot units are fundamentally different from the story representations that preceded them because they focus on the affect states of characters and the tensions between them as the driving force behind interesting and cohesive stories. Plot units were used in narrative summarization studies, both in computer science and psychology (Lehnert et al., 1981), but previous computational models of plot units relied on tremendous amounts of manual knowledge engineering.

The last few decades have seen tremendous advances in NLP and the emergence of many resources that could be useful for plot unit analysis. So we embarked on a project to see whether plot unit representations can be generated automatically using current NLP technology. We created a system called AESOP that uses a variety of resources to identify words that correspond to positive, negative, and mental *affect states*. AESOP uses *affect projection rules* to map the affect states onto the characters in the story based on verb argument structure. Additionally, affect states are inferred based on syntactic properties, and causal and cross-character links are created using simple heuristics.

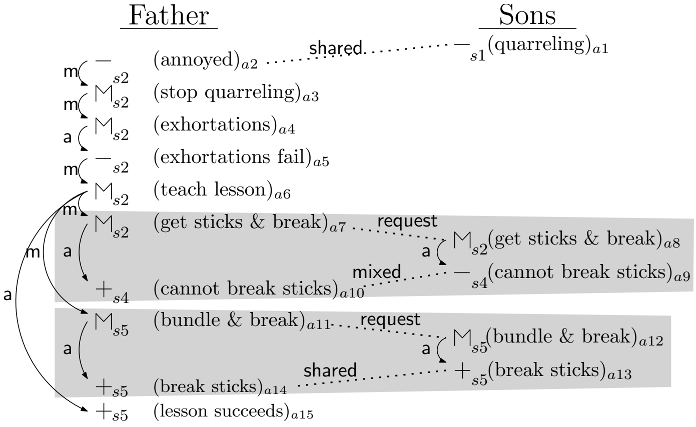
Affect states often arise from actions that produce good or bad states for the character that is acted upon. For example, *"the cat ate the mouse"* produces a negative state for the mouse because being eaten is bad. Similarly, *"the man fed the dog"* produces a positive state for the dog because being fed is generally good. Knowledge about the effects of actions (i.e., state changes) on patients is not readily available in existing semantic resources. We create a new type of lexicon consisting of *patient polarity verbs (PPVs)* that impart positive or negative states on their patients. These verbs reflect world knowledge about desirable/undesirable states for animate beings; for example, being *fed*, *paid* or *adopted* are generally desirable states, while being *eaten*, *chased* or *hospitalized* are generally undesirable states.

We automatically generate a lexicon of "patient polarity verbs" from a Web corpus using two tech-

The Father and His Sons

(s1) A father had a family of sons who were perpetually quarreling among themselves. (s2) When he failed to heal their disputes by his exhortations, he determined to give them a practical illustration of the evils of disunion; and for this purpose he one day told them to bring him a bundle of sticks. (s3) When they had done so, he placed the faggot into the hands of each of them in succession, and ordered them to break it in pieces. (s4) They tried with all their strength, and were not able to do it. (s5) He next opened the faggot, took the sticks separately, one by one, and again put them into his sons' hands, upon which they broke them easily. (s6) He then addressed them in these words: "My sons, if you are of one mind, and unite to assist each other, you will be as this faggot, uninjured by all the attempts of your enemies; but if you are divided among yourselves, you will be broken as easily as these sticks."

(a) "Father and Sons" Fable



(b) Plot Unit Analysis for "Father and Sons" Fable

Figure 1: Sample Fable and Plot Unit Representation

niques: patterns that identify co-occurrence with stereotypically evil or kind agents, and a bootstrapping algorithm that learns from conjunctions of verbs. We evaluate the plot unit representations produced by our system on a small collection of fables.

2 Overview of Plot Units

Plot unit structures consist of *affect states* for each character, and links defining the relationships between them. Plot units include three types of affect states: positive (+), negative (-), and mental (M). Affect states can be connected by *causal links* and *cross-character* links, which explain how the narrative hangs together. Causal links exist between affect states for the same character and have four types: motivation (m), actualization (a), termination (t) and equivalence (e). Cross-character links indicate that a single event affects multiple characters. For instance, if one character *requests* something of another, then each character is assigned an M state and a cross-character link connects the states.

To see a concrete example of a plot unit representation, a short fable, "The Father and His Sons," is shown in Figure 1(a) and our annotation of its plot unit structure is shown in Figure 1(b). In this fable, there are two characters, the "Father" and (collectively) the "Sons", who go through a series of affect states depicted chronologically in the two columns.

The first affect state ($a1$) is produced from sentence #1 ($s1$) and is a negative state for the sons because they are quarreling. This state is *shared* by the father (via a cross-character link) who has a negative annoyance state ($a2$). The father decides that he wants to stop the sons from quarreling, which is a mental event ($a3$). The causal link from $a2$ to $a3$ with an m label indicates that his annoyed state "motivated" this decision. His first attempt is by exhortations ($a4$). The first M ($a3$) is connected to the second M ($a4$) with an m (motivation) link, which represents subgoaling. The father's overall goal is to stop the quarreling ($a3$), and to do so he creates a subgoal of exhorting the sons to stop ($a4$). The exhortations fail, which produces a negative state ($a5$) for the father. The a causal link indicates an "actualization", representing the failure of his plan ($a4$).

This failure motivates a new subgoal: teach the sons a lesson ($a6$). At a high level, this subgoal has two parts, indicated by the two gray regions ($a7 - a10$ and $a11 - a14$). The first gray region begins with a cross-character link (M to M), which indicates a request (in this case, to break a bundle of sticks). The sons fail at this, which upsets them ($a9$) but pleases the father ($a10$). The second gray region depicts the second part of the father's subgoal; he makes a second request ($a11$ to $a12$) to separate the bundle and break the sticks, which the sons successfully do, making them happy ($a13$) and the father happy ($a14$) as well. This latter structure (the second gray region) is an HONORED REQUEST plot unit structure. At the end, the father's plan succeeds ($a15$) which is an actualization (a link) of his goal to teach the sons a lesson ($a6$).

3 Where Do Affect States Come From?

We briefly overview the variety of situations that can be represented by affect states in plot units.

Direct Expressions of Emotion: Affect states can correspond to positive/negative emotional states, as have been studied in the realm of sentiment analysis. For example, *“Max was disappointed”* produces a negative affect state for Max, and *“Max was pleased”* produces a positive affect state for Max.

Situational Affect States: Positive and negative affect states can represent good and bad situational states that characters find themselves in. These states do not represent emotion, but indicate whether a situation (state) is good or bad for a character based on world knowledge. e.g., *“The wolf had a bone stuck in his throat.”* produces a negative affect state for the wolf. Similarly, *“The woman recovered her sight.”* produces a positive affect state for the woman.

Plans and Goals: The existence of a plan or goal is represented as a mental state (M). Plans and goals can be difficult to detect automatically and can be revealed in many ways, such as:

- **Direct expressions of plans/goals:** a plan/goal may be explicitly stated (e.g., *“John wants food”*).
- **Speech acts:** a plan or goal may be revealed through a speech act. For example, *“the wolf asked an eagle to extract the bone”* is a directive speech act that indicates the wolf’s plan to resolve its negative state (having a bone stuck). This example illustrates how a negative state (bone stuck) can motivate a mental state (plan). When a speech act involves multiple characters, it produces multiple mental states.
- **Inferred plans/goals:** plans and goals are sometimes inferred from actions. e.g., *“the lion hunted deer”* implies that the lion has a plan to obtain food. Similarly, *“the serpent spat poison at John”* implies that the serpent wants to kill John.
- **Plan/Goal completion:** Plans and goals produce +/- affect states when they succeed or fail. For example, if the eagle successfully extracts the bone from the wolf’s throat, then both the wolf and the eagle will have positive affect states because both were successful in their respective goals.

We observed that situational and plan/goal states often originate from an action. When a character is acted upon (the *patient* of a verb), then the character may be in a positive or negative state depending upon whether the action was good or bad for them based on world knowledge. For example, being *fed*, *paid* or *adopted* is generally desirable, but being *chased*, *eaten*, or *hospitalized* is usually undesirable. Consequently, we decided to create a lexicon of *patient polarity verbs* that produce positive or negative states for their patients. In Section 4.2, we present two methods for automatically harvesting these verbs from a Web corpus.

4 AESOP: Automatically Generating Plot Unit Representations

Our system, AESOP, automatically creates plot unit representations for narrative text. AESOP has four main steps: affect state recognition, character identification, affect state projection, and link creation. During affect state recognition, AESOP identifies words that may be associated with positive, negative, and mental states. AESOP then identifies the main characters in the story and applies *affect projection rules* to map the affect states onto these characters. During this process, some additional affect states are inferred based on verb argument structure. Finally, AESOP creates cross-character links and causal links between affect states. We also present two corpus-based methods to automatically produce a new resource for affect state recognition: a *patient polarity verb lexicon.*

4.1 Plot Unit Creation

4.1.1 Recognizing Affect States

The basic building blocks of plot units are *affect states* which come in three flavors: positive, negative, and mental. In recent years, many publicly available resources have been created for sentiment analysis and other types of semantic knowledge. We considered a wide variety of resources and ultimately decided to experiment with five resources that most closely matched our needs:

- FrameNet (Baker et al., 1998): We manually identified 87 frame classes that seem to be associated with affect: 43 mental classes (e.g., COMMUNICATION and NEEDING), 22 positive classes (e.g., ACCOMPLISHMENT and SUPPORTING), and 22 negative classes (e.g., CAUSE HARM and PROHIBIT-

ING). We use the verbs listed for these classes to produce M, +, and - affect states.

- MPQA Lexicon (Wilson et al., 2005b): We used the words listed as having positive or negative sentiment polarity to produce +/- states, when they occur with the designated part-of-speech.

- OpinionFinder (Wilson et al., 2005a) (Version 1.4) : We used the +/- labels assigned by its contextual polarity classifier (Wilson et al., 2005b) to create +/- states and the MPQASD tags produced by its Direct Subjective and Speech Event Identifier (Choi et al., 2006) to produce mental (M) states.

- Semantic Orientation Lexicon (Takamura et al., 2005): We used the words listed as having positive or negative polarity to produce +/- affect states, when they occur with the designated part-of-speech.

- Speech Act Verbs: We used 228 speech act verbs from (Wierzbicka, 1987) to produce M states.

4.1.2 Identifying the Characters

For the purposes of this work, we made two simplifying assumptions: (1) There are only two characters per fable[1], and (2) Both characters are mentioned in the fable's title. The problem of coreference resolution for fables is somewhat different than for other genres, primarily because the characters are often animals (e.g., *he=owl*). So we hand-crafted a simple rule-based coreference system. First, we apply heuristics to determine number and gender based on word lists, WordNet (Miller, 1990) and part-of-speech tags. If no determination of a character's gender or number can be made, we employ a process of elimination. Given the two character assumption, if one character is known to be male, but there are female pronouns in the fable, then the other character is assumed to be female. The same is done for number agreement. Finally, if there is only one character between a pronoun and the beginning of a document, then we resolve the pronoun with that character and the character assumes the gender and number of the pronoun. Lastly, WordNet provides some additional resolutions by exploiting hypernym relations, for instance, linking *peasant* with *man*.

4.1.3 Mapping Affect States onto Characters

Plot unit representations are not just a set of affect states, but they are structures that capture the chronological ordering of states for each character as the narrative progresses. Consequently, every affect state needs to be attributed to a character. Since most plots revolve around events, we use verb argument structure as the primary means for projecting affect states onto characters.

We developed four *affect projection rules* that orchestrate how affect states are assigned to the characters. We used the Sundance parser (Riloff and Phillips, 2004) to produce a shallow parse of each sentence, which includes syntactic chunking, clause segmentation, and active/passive voice recognition. We normalized the verb phrases with respect to active/passive voice to simplify the rules. We made the assumption that the Subject of the VP is its AGENT and the Direct Object of the VP is its PATIENT.[2] The rules only project affect states onto AGENTS and PATIENTS that refer to a character in the story. The four projection rules are presented below.

1. AGENT **VP** : This rule applies when the VP has no PATIENT or the PATIENT corefers with the AGENT. All affect tags assigned to the VP are projected onto the AGENT. Example: *"Mary **laughed** (+)"* projects a + affect state onto Mary.

2. **VP** PATIENT : This rule applies when the VP has no agent, which is common in passive voice constructions. All affect tags assigned to the VP are projected onto the PATIENT. Example: *"John was **rewarded** (+)*, projects a + affect state onto John.

3. AGENT **VP** PATIENT : This rules applies when both an AGENT and PATIENT are present, do not corefer, and at least one of them is a character. If the PATIENT is a character, then all affect tags associated with the VP are projected onto the PATIENT. If the AGENT is a character and the VP has an M tag, then we also project an M tag onto the AGENT (representing a shared, cross-character mental state).

4. AGENT **VERB1** to **VERB2** PATIENT : This rule has two cases: (a) If the AGENT and PATIENT refer to the same character, then we apply Rule #1. Example: *"Bo decided to teach himself..."* (b) If the AGENT and PATIENT are different, then we apply Rule #1 to **VERB1** and Rule #2 to **VERB2**.

Finally, if an adverb or adjectival phrase has affect, then that affect is mapped onto the preceding VP and the rules above are applied. For all of the

[1] We only selected fables that had two main characters.

[2] This is not always correct, but worked ok in our fables.

rules, if a clause contains a negation word, then we flip the polarity of all words in that clause.

4.1.4 Inferring Affect States

Recognizing plans and goals depends on world knowledge and inference, and is beyond the scope of this paper. However, we identified two cases where affect states often can be inferred based on syntactic properties. The first case involves verb phrases (VPs) that have both an AGENT and PATIENT, which corresponds to projection rule #3. If the VP has polarity, then rule #3 assigns that polarity to the PATIENT, not the AGENT. For example, *"John killed Paul"* imparts negative polarity on Paul, but not necessarily on John. Unless we are told otherwise, one assumes that John *intentionally* killed Paul, and so in a sense, John accomplished his goal. Consequently, this action should produce a positive affect state for John. We capture this notion of accomplishment as a side effect of projection rule #3: if the VP has +/- polarity, then we produce an *inferred positive state* for the AGENT.

The second case involves infinitive verb phrases of the form: "AGENT VERB1 TO VERB2 PATIENT" (e.g., *"Susan tried to warn Mary"*). The infinitive VP construction suggests that the AGENT has a goal or plan that is being put into motion (e.g., *tried to, wanted to, attempted to, hoped to*, etc.). To capture this intuition, in rule #4 if VERB1 does not already have an affect state assigned to it then we produce an *inferred mental state* for the AGENT.

4.1.5 Causal and Cross-Character Links

Our research is focused primarily on creating affect states for characters, but plot unit structures also include *cross-character links* to connect states that are shared across characters and *causal links* between states for a single character. As an initial attempt to create complete plot units, AESOP produces links using simple heuristics. A cross-character link is created when two characters in a clause have affect states that originated from the same word. A causal link is created between each pair of (chronologically) consecutive affect states for the same character. Currently, AESOP only produces forward causal links (*motivation (m)*, *actualization (a)*) and does not produce backward causal links (*equivalence (e)*, *termination (t)*). For forward links, the causal syntax only allows for five cases: $M \xrightarrow{m} M$, $+ \xrightarrow{m} M$, $- \xrightarrow{m} M$, $M \xrightarrow{a} +$, $M \xrightarrow{a} -$. So when AESOP produces a causal link between two affect states, the order and types of the two states uniquely determine which label it gets (m or a).

4.2 Generating PPV Lexicons

During the course of this research, we identified a gap in currently available knowledge: we are not aware of existing resources that identify verbs which produce a desirable/undesirable state for their patients even though the verb itself does not carry polarity. For example, the verb *eat* describes an action that is generally neutral, but being eaten is clearly an undesirable state. Similarly, the verb *fed* does not have polarity, but being fed is a desirable state for the patient. In the following sections, we try to fill this gap by using corpus-based techniques to automatically acquire a *Patient Polarity Verb (PPV) Lexicon*.

4.2.1 PPV Harvesting with Evil/Kind Agents

The key idea behind our first approach is to identify verbs that frequently occur with evil or kind agents. Our intuition was that an "evil" agent will typically perform actions that are bad for the patient, while a "kind" agent will typically perform actions that are good for the patient.

We manually identified 40 stereotypically evil agent words, such as *monster*, *villain*, *terrorist*, and *murderer*, and 40 stereotypically kind agent words, such as *hero*, *angel*, *benefactor*, and *rescuer*. We searched the Google Web $1T$ N-gram corpus to identify verbs that co-occur with these words as probable agents. For each agent term, we applied the pattern "* by [a,an,the] AGENT" and extracted the matching N-grams. Then we applied a part-of-speech tagger to each N-gram and saved the words that were tagged as verbs (i.e., the words in the * position).[3] This process produced 811 negative (evil agent) PPVs and 1362 positive (kind agent) PPVs.

4.2.2 PPV Bootstrapping over Conjunctions

Our second approach for acquiring PPVs is based on an observation from sentiment analysis research that conjoined adjectives typically have the same polarity (e.g. (Hatzivassiloglou and McKeown, 1997)).

[3] The POS tagging quality is undoubtedly lower than if tagging complete sentences but it seemed reasonable.

Our hypothesis was that conjoined verbs often share the same polarity as well (e.g., *"abducted and killed"* or *"rescued and rehabilitated"*). We exploit this idea inside a bootstrapping algorithm to iteratively learn verbs that co-occur in conjunctions.

Bootstrapping begins with 10 negative and 10 positive PPV seeds. First, we extracted triples of the form *"w1 and w2"* from the Google Web $1T$ N-gram corpus that had frequency ≥ 100 and were lower case. We separated each conjunction into two parts: a primary VERB ("$w1$") and a CONTEXT ("$and\ w2$"), and created a copy of the conjunction with the roles of $w1$ and $w2$ reversed. For example, *"rescued and adopted"* produces:

VERB="rescued" CONTEXT="and adopted"
VERB="adopted" CONTEXT="and rescued"

Next, we applied the Basilisk bootstrapping algorithm (Thelen and Riloff, 2002) to learn PPVs. Basilisk identifies semantically similar words based on their co-occurrence with seeds in contextual patterns. Basilisk was originally designed for semantic class induction using lexico-syntactic patterns, but has also been used to learn subjective and objective nouns (Riloff et al., 2003).

Basilisk first identifies the pattern contexts that are most strongly associated with the seed words. Words that occur in those contexts are labeled as *candidates* and scored based on the strength of their contexts. The top 5 candidates are selected and the bootstrapping process repeats. Basilisk produces a *lexicon* of learned words as well as a ranked list of pattern contexts. Since we bootstrapped over verb conjunctions, we also extracted new PPVs from the contexts. We ran the bootstrapping process to create a lexicon of 500 words, and we collected verbs from the top 500 contexts as well.

5 Evaluation

Plot unit analysis of narrative text is enormously complex – the idea of creating gold standard plot unit annotations seemed like a monumental task. So we began with relatively simple and constrained texts that seemed appropriate: fables. Fables have two desirable attributes: (1) they have a small cast of characters, and (2) they typically revolve around a moral, which is exemplified by a short and concise plot. Even so, fables are challenging for NLP due to anthropomorphic characters, flowery language, and sometimes archaic vocabulary.

We collected 34 Aesop's fables from a web site[4], choosing fables that have a true plot (some only contain quotes) and exactly two characters. We divided them into a development set of 11 stories, a tuning set of 8 stories, and a test set of 15 stories.

Creating a gold standard was itself a substantial undertaking, and training non-experts to produce them did not seem feasible in the short term. So the authors discussed and iteratively refined manual annotations for the development and tuning sets until we produced similar results and had a common understanding of the task. Then two authors independently created annotations for the test set, and a third author adjudicated the differences.

5.1 Evaluation Procedure

For evaluation, we used recall (R), precision (P), and F-measure (F). In our gold standard, each affect state is annotated with the set of clauses that could legitimately produce it. In most cases (75%), we were able to ascribe the existence of a state to precisely one clause. During evaluation, the system-produced affect states must be generated from the correct clause. However, for affect states that could be ascribed to multiple clauses in a sentence, the evaluation was done at the sentence level. In this case, the system-produced affect state must come from the sentence that contains one of those clauses.

Coreference resolution is far from perfect, so we created gold standard coreference annotations for our fables and used them for most of our experiments. This allowed us to evaluate our approach without coreference mistakes factoring in. In Section 5.5, we re-evaluate our final results using automatic coreference resolution.

5.2 Evaluation of Affect States using External Resources

Our first set of experiments evaluates the quality of the affect states produced by AESOP using only the external resources. The top half of Table 1 shows the results for each resource independently. FrameNet produced the best results, yielding much higher recall than any other resource. The bottom half of Ta-

[4] www.pacificnet.net/~johnr/aesop/

Affect State	**M** (59)			**+** (47)			**-** (37)			**All** (143)		
Resource(s)	*R*	*P*	*F*	*R*	*P*	*F*	*R*	*P*	*F*	*R*	*P*	*F*
FrameNet	.49	.51	.50	.17	.57	.26	.14	.42	.21	.29	.51	.37
MPQA Lexicon	.07	.50	.12	.21	.24	.22	.22	.38	.28	.15	.31	.20
OpinionFinder	.42	.40	.41	.00	.00	.00	.03	.17	.05	.18	.35	.24
Semantic Orientation Lexicon	.07	.44	.12	.17	.40	.24	.08	.38	.13	.10	.41	.16
Speech Act Verbs	.36	.53	.43	.00	.00	.00	.00	.00	.00	.15	**.53**	.23
FrameNet+MPQA Lexicon	.44	.52	.48	.30	.28	.29	.27	.38	.32	**.35**	.40	.37
FrameNet+OpinionFinder	.53	.39	.45	.17	.38	.23	.16	.33	.22	.31	.38	.34
FrameNet+Semantic Orientation Lexicon	.49	.51	.50	.26	.36	.30	.22	.42	.29	.34	.45	**.39**
FrameNet+Speech Act Verbs	.51	.48	.49	.17	.57	.26	.14	.42	.21	.30	.49	.37

Table 1: Evaluation results for AESOP using external resources. The # in parentheses is the # of gold affect states.

Affect State	**M** (59)			**+** (47)			**-** (37)			**All** (143)		
Resource(s)	*R*	*P*	*F*	*R*	*P*	*F*	*R*	*P*	*F*	*R*	*P*	*F*
- Evil Agent PPVs	.07	.50	.12	.21	.40	.28	.46	.46	.46	.22	.44	.29
- Neg Basilisk PPVs	.07	.44	.12	.11	.45	.18	.24	.45	.31	.13	.45	.20
- Evil Agent and Neg Basilisk PPVs	.05	.43	.09	.21	.38	.27	.46	.40	.43	.21	.39	.27
+ Kind Agent PPVs (θ>1)	.03	.33	.06	.28	.17	.21	.00	.00	.00	.10	.19	.13
+ Pos Basilisk PPVs	.08	.56	.14	.02	.12	.03	.03	1.00	.06	.05	.39	.09
FrameNet+SOLex+EvilAgentPPVs	.49	.54	.51	.30	.38	.34	.46	.42	.44	.42	.46	.44
FrameNet+EvilAgentPPVs	.49	.54	.51	.28	.45	.35	.46	.46	.46	.41	.49	**.45**
FrameNet+EvilAgentPPVs+PosBasiliskPPVs	.49	.53	.51	.30	.41	.35	.49	.49	.49	**.43**	.48	**.45**

Table 2: Evaluation results for AESOP with PPVs. The # in parentheses is the # of gold affect states.

ble 1 shows the results when combining FrameNet with other resources. In terms of F score, the only additive benefit came from the Semantic Orientation Lexicon, which produced a better balance of recall and precision and an F score gain of +2.

5.3 Evaluation of Affect States using PPVs

Our second set of experiments evaluates the quality of the automatically generated PPV lexicons. The top portion of Table 2 shows the results for the negative PPVs. The PPVs harvested by the Evil Agent patterns produced the best results, yielding recall and precision of .46 for negative states. Note that M and + states are also generated from the negative PPVs because they are inferred during affect projection (Section 4.1.4). The polarity of a negative PPV can also be flipped by negation to produce a + state.

Basilisk's negative PPVs achieved similar precision but lower recall. We see no additional recall and some precision loss when the Evil Agent and Basilisk PPV lists are combined. The precision drop is likely due to redundancy, which creates spurious affect states. If two different words have negative polarity but refer to the same event, then only one negative affect state should be generated. But AESOP will generate two affect states, so one will be spurious.

The middle section of Table 2 shows the results for the positive PPVs. Both positive PPV lexicons were of dubious quality, so we tried to extract a high-quality subset of each list. For the Kind Agent PPVs, we computed the ratio of the frequency of the verb with Evil Agents versus Kind Agents and only saved verbs with an Evil:Kind ratio (θ) > 1, which yielded 1203 PPVs. For the positive Basilisk PPVs, we used only the top 100 lexicon and top 100 context verbs, which yielded 164 unique verbs. The positive PPVs did generate several correct affect states (including a - state when a positive PPV was negated), but also many spurious states.

The bottom section of Table 2 shows the impact of the learned PPVs when combined with FrameNet and the Semantic Orientation Lexicon (SOLex). Adding the Evil Agent PPVs improved AESOP's F score from 39% to 44%, mainly due to a +8 recall gain. The recall of the - states increased from 22% to 46% with no loss of precision. Interestingly, if we remove SOLex and use only FrameNet with our PPVs, precision increases from 46% to 49% and recall only drops by -1. Finally, the last row of Table

2 shows that adding Basilisk's positive PPVs produces a small recall boost (+2) with a slight drop in precision (-1).

Evaluating the impact of PPVs on plot unit structures is an indirect way of assessing their quality because creating plot units involves many steps. Also, our test set is small so many verbs will never appear. To directly measure the quality of our PPVs, we recruited 3 people to manually review them. We developed annotation guidelines that instructed each annotator to judge whether a verb is generally *good* or *bad* for its patient, assuming the patient is animate. They assigned each verb to one of 6 categories: × (not a verb), 2 (always good), 1 (usually good), 0 (neutral, mixed, or requires inanimate patient), -1 (usually bad), -2 (always bad). Each annotator labeled 250 words: 50 words randomly sampled from each of our 4 PPV lexicons[5] (Evil Agent PPVs, Kind Agent PPVs, Positive Basilisk PPVs, and Negative Basilisk PPVs) plus 50 verbs labeled as neutral in the MPQA lexicon.

First, we measured agreement based on three groupings: *negative* (-2 and -1), *neutral* (0), or *positive* (1 and 2). We computed κ scores to measure inter-annotator agreement for each pair of annotators.[6], but the κ scores were relatively low because the annotators had trouble distinguishing the positive cases from the neutral ones. So we re-computed agreement using two groupings: *negative* (-2 and -1) and *not-negative* (0 through 2), and obtained κ scores of .69, .71, and .74. We concluded that people largely agree on whether a verb is bad for the patient, but they do not necessarily agree if a verb is good for the patient. One possible explanation is that many "bad" verbs represent physical harm or danger: these verbs are both plentiful and easy to recognize. In contrast, "good" verbs are often more abstract and open to interpretation (e.g., is being "envied" or "feared" a good thing?).

We used the labels produced by the two annotators with the highest κ score to measure the accuracy of our PPVs. Both the Evil Agent and Negative Basilisk PPVs were judged to be 72.5% accurate, averaged over the judges. The Kind Agent PPVs were only about 39% accurate, while the Positive Basilisk PPVs were nearly 50% accurate. These results are consistent with our impressions that the negative PPVs are of relatively high quality, while the positive PPVs are mixed. Some examples of learned PPVs that were not present in our other resources are:

- : censor, chase, fire, orphan, paralyze, scare, sue
+ : accommodate, harbor, nurse, obey, respect, value

5.4 Evaluation of Links

We represented each link as a 5-tuple $\langle$*src-clause*, *src-state*, *tgt-clause*, *tgt-state*, *link-type*$\rangle$, where source/target denotes the direction of the link, the source/target-states are the affect state type (+,-,M) and *link-type* is one of 3 types: actualization (a), motivation (m), or cross-character (xchar). A system-produced link is considered correct if *all* 5 elements of the tuple match the human annotation.

	Gold Aff States			*System Aff States*		
Links	*R*	*P*	*F*	*R*	*P*	*F*
xchar (56)	.79	.85	.82	.18	.43	.25
a (51)	.90	.94	.92	.04	.07	.05
m (26)	1.0	.57	.72	.15	.10	.12

Table 3: Link results; parentheses show # of gold links.

The second column of Table 3 shows the performance of AESOP when using gold standard affect states. Our simple heuristics for creating links work surprisingly well for xchar and a links when given perfect affect states. However, these heuristics produce relatively low precision for m links, albeit with 100% recall. This reveals that m links primarily do connect adjacent states, but we need to be more discriminating when connecting them. The third column of Table 3 shows the results when using system-generated affect states. We see that performance is much lower. This is not particularly surprising, since AESOP's F-score is 45%, so over half of the individual states are wrong, which means that less than a quarter of the pairs are correct. From that perspective, the xchar link performance is reasonable, but the causal a and m links need improvement.

5.5 Analysis

We performed additional experiments to evaluate some assumptions and components. First, we created a *Baseline* system that is identical to AESOP

[5] The top-ranked Evil/Kind Agent PPV lists ($\theta > 1$) which yields 1203 kind PPVs, and 477 evil PPVs, the top 164 positive Basilisk verbs, and the 678 (unique) negative Basilisk verbs.

[6] We discarded words labeled as not a verb.

except that it does not use the affect projection rules. Instead, it naively projects every affect state in a clause onto every character in that clause. The first two rows of the table below show that AESOP's precision is double the Baseline, with nearly the same recall. This illustrates the importance of the projection rules for mapping affect states onto characters.

	R	*P*	*F*
Baseline	.44	.24	.31
AESOP, gold coref	.43	.48	.45
AESOP, gold coref, infstates	.39	.48	.43
AESOP, auto coref, infstates	.24	.56	.34

Our gold standard includes *pure inference affect states* that are critical to the plot unit structure but come from world knowledge outside the story itself. Of 157 affect states in our test set, 14 were pure inference states. We ignored these states in our previous experiments because our system has no way to generate them. The third row of the table shows that including them lowers recall by -4. Generating pure inferences is an interesting challenge, but they seem to be a relatively small part of the problem.

The last row of the table shows AESOP's performance when we use our automated coreference resolver (Section 4.1.2) instead of gold standard coreference annotations. We see a -15 recall drop coupled with a +8 precision gain. We were initially puzzled by the precision gain but believe that it is primarily due to the handling of quotations. Our gold standard includes annotations for characters mentioned in quotations, but our automated coreference resolver ignores quotations. Most fables end with a moral, which is often a quote that may not mention the plot. Consequently, AESOP generates more spurious affect states from the quotations when using the gold standard annotations.

6 Related Work and Conclusions

Our research is the first effort to fully automate the creation of plot unit structures. Other preliminary work has begun to look at plot unit modelling for single character stories (Appling and Riedl, 2009). More generally, our work is related to research in narrative story understanding (e.g., (Elson and McKeown, 2009)), automatic affect state analysis (Alm, 2009), and automated learning of scripts (Schank and Abelson, 1977) and other conceptual knowledge structures (e.g., (Mooney and DeJong, 1985; Fujiki et al., 2003; Chambers and Jurafsky, 2008; Chambers and Jurafsky, 2009; Kasch and Oates, 2010)). Our work benefitted from prior research in creating semantic resources such as FrameNet (Baker et al., 1998) and sentiment lexicons and classifiers (e.g., (Takamura et al., 2005; Wilson et al., 2005b; Choi et al., 2006)). We showed that *affect projection rules* can effectively assign affect states to characters. This task is similar to, but not the same as, associating opinion words with their targets or topics (Kim and Hovy, 2006; Stoyanov and Cardie, 2008). Some aspects of affect state identification are closely related to Hopper and Thompson's (1980) theory of transitivity. In particular, their notions of *aspect* (has an action completed?), benefit and harm (how much does an object gain/lose from an action?) and volition (did the subject make a conscious choice to act?).

AESOP produces affect states with an F score of 45%. Identifying *positive* states appears to be more difficult than negative or mental states. Our system's biggest shortcoming currently seems to hinge around identifying plans and goals. This includes the M affect states that initiate plans, the +/- completion states, as well as their corresponding links. We suspect that the relatively low recall on positive affect states is due to our inability to accurately identify successful plan completions. Finally, these results are based on fables; plot unit analysis of other types of texts will pose additional challenges.

Acknowledgments

The authors gratefully acknowledge the support of Department of Homeland Security Grant N0014-07-1-0152, NSF grant IIS-0712764, and the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0172. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the DARPA, AFRL, or the U.S. government. Thanks to Peter Jensen, Emily Schlichter, and Clay Templeton for PPV annotations, Nathan Gilbert for help with the coreference resolver, and the anonymous reviewers for many helpful comments.

References

Cecilia Ovesdotter Alm. 2009. *Affect in Text and Speech*. VDM Verlag Dr. Mller.

D. Scott Appling and Mark O. Riedl. 2009. Representations for learning to summarize plots. In *Proceedings of the AAAI Spring Symposium on Intelligent Narrative Technologies II*.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *In Proceedings of COLING/ACL*, pages 86–90.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of the Association for Computational Linguistics*.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Association for Computational Linguistics*.

Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Morristown, NJ, USA. Association for Computational Linguistics.

David Elson and Kathleen McKeown. 2009. Extending and evaluating a platform for story understanding. In *Proceedings of the AAAI 2009 Spring Symposium on Intelligent Narrative Technologies II*.

Toshiaki Fujiki, Hidetsugu Nanba, and Manabu Okumura. 2003. Automatic acquisition of script knowledge from a text collection. In *Proceedings of the European Association for Computational Linguistics*.

Vasileios Hatzivassiloglou and Kathy McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 174–181, Madrid, Spain.

Paul J. Hopper and Sandra A. Thompson. 1980. Transitivity in grammar and discourse. *Language*, 56:251299.

Niels Kasch and Tim Oates. 2010. Mining script-like structures from the web. In *NAACL-10 Workshop on Formalisms and Methodology for Learning by Reading (FAM-LbR)*.

S. Kim and E. Hovy. 2006. Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. In *Proceedings of ACL/COLING Workshop on Sentiment and Subjectivity in Text*.

W. Lehnert, J. Black, and B. Reiser. 1981. Summarizing Narratives. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*.

W. G. Lehnert. 1981. Plot Units and Narrative Summarization. *Cognitive Science*, 5(4):293–331.

G. Miller. 1990. Wordnet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4).

Raymond Mooney and Gerald DeJong. 1985. Learning Schemata for Natural Language Processing. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 681–687.

E. Riloff and W. Phillips. 2004. An Introduction to the Sundance and AutoSlog Systems. Technical Report UUCS-04-015, School of Computing, University of Utah.

E. Riloff, J. Wiebe, and T. Wilson. 2003. Learning Subjective Nouns using Extraction Pattern Bootstrapping. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, pages 25–32.

Roger C. Schank and Robert P. Abelson. 1977. *Scripts, plans, goals and understanding*. Lawrence Erlbaum.

V. Stoyanov and C. Cardie. 2008. Topic Identification for Fine-Grained Opinion Analysis. In *Conference on Computational Linguistics (COLING 2008)*.

Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*.

M. Thelen and E. Riloff. 2002. A Bootstrapping Method for Learning Semantic Lexicons Using Extraction Pattern Contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 214–221.

A. Wierzbicka. 1987. *English speech act verbs: a semantic dictionary*. Academic Press, Sydney, Orlando.

T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. 2005a. OpinionFinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005b. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354. Association for Computational Linguistics.