Representations for Learning to Summarize Plots

D. Scott Appling and Mark O. Riedl
School of Interactive Computing, Georgia Institute of Technology
85 Fifth Street NW, Atlanta, Georgia 30308, USA
darren.scott.appling@gatech.edu; riedl@gatech.edu

Abstract

Stories can encapsulate complexity, subtlety, and nuance: all of which are implicitly contained in narrative and reasoned about automatically through the mental processes that come naturally to humans. For example, humans can package complicated plots into a relatively small set of well-recognized and meaningful linguistic terms. This summarization ability though has not been available to systems that deal with narrative and would be important in creating higher quality systems. In this paper, we describe preliminary work towards a machine learning model of plot summarization using *conditional random fields* and describe our own feature functions inspired by cognitive theories of narrative reasoning. Our approach allows us to learn summarization models of single character event driven narratives and automatically summarize new narratives later on.

Introduction

Storytelling is a pervasive part of human culture. We use stories to communicate, entertain, and educate. Stories are more than just interesting artifacts that are created, narratives are a fundamental means by which humans organize, understand, and explain the world (Bruner 1990). However, due to the complexity, subtlety, and nuance that can be expressed through stories, automated reasoning about narrative cannot yet perform some of the mental processes that come naturally to humans. For example, story understanding systems such as (Cullingford 1981; Wilensky 1981; Lehnert et al. 1983; Ram 1994; Mueller 2004) demonstrate their abilities by inferring answers to questions posed by a human user. Lehnert (1982) notes that when people think and communicate about stories they have read or seen, they frequently summarize the plots. Lehnert attempts to enumerate many of the terms that humans use to summarize plots – that is, terms that package up complicated narrative structures for quick and easy dissemination and digestion.

Lehnert (1982) provides a partial set of candidate terms that describe commonly used plot structures, such as “killing two birds with one stone,” “intentional problem resolution,” and “hidden blessing.” However, Lehnert does not provide an algorithm for summarizing plots. One of the reasons plot summarization is hard is because story plots – which can be summarized simply – can manifest themselves in nearly infinite ways. In this paper, we describe preliminary work towards a machine learning model of plot summarization. Our approach is to use *conditional random fields* to learn to label parts of narratives with plot unit descriptors from a human-annotated corpus. Our approach to plot summarization will enable computational systems to reason about stories in terms that are familiar to humans. Additionally, by learning a model, we may in the future be able to create more sophisticated story evaluation functions that can be used to guide story generation systems.

Plot Units

Lehnert (1982) posits that narratives are summarized by readers according to affect-state patterns. That is, a mental model of a narrative consists of – physical and mental character acts – and state propositions. These states and events are recalled according to their affect: positive events for the acting character (+), negative events for the acting character (-), or neutral mental states (M). States and events are linked according to the way they relate to one another:

- Mental states can motivate events or other mental states (m)
- Mental states can be actualized by events (a)
- Events can terminate mental states or other events (t)
- Mental states and events can be equivalent (e)

Particular patterns of mental states and events, according to the way they are linked, are called primitive plot units. Figure 1 shows an example of the *hidden blessing* primitive plot unit, in which a event, initially regarded with negative affect by a character, is later regarded with positive affect.

Primitive plot units can be used to build complex plot units. Figure 1 shows an example of the *intentional problem resolution* complex plot unit, which consists of a *problem* primitive plot unit (a negative event motivating a mental state), a *resolution* primitive plot unit (a positive event terminating a negative event), and a *success* primitive plot unit (a mental state actualizing a positive event). A specific instantiation of an intentional problem resolution would be the following events: a character wrecks his car – a negative event – which motivates the formation of a goal to replace the car

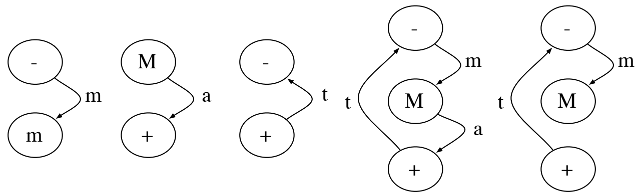

Figure 1: Examples of plot units (left to right): (a) *problem*, (b) *success*, (c) *resolution*, (d) *intentional problem resolution*, and (e) *fortuitous problem resolution*.

– a mental state – which is actualized by buying of a new car – a positive event. The buying of the new car terminates the wreck event because it creates closure. However, suppose instead of buying the car, the character were given a new car by the insurance company. In this case, the characters goal is satisfied by the actions of an external agent and, although the events would be very similar to the previous example, there would not be an actualization link from the mental state to the positive state. In this case, a better summarization would be *fortuitous problem resolution*, a combination of *problem* and *resolution* primitive units.

Learning to Summarize Plots

We are developing a system to learn from readers how to automatically reason about plot unit summarization involved in single character narratives. These stories are aggregations of temporally ordered *events*. There are however events that different readers might leave out if they were summarizing a story to another person. Taking into account this nature, and that of plot units i.e. their implicit summarization, it inherently follows that not all events of a story should necessarily be kept in a summary. That is, only the most important events would be kept to preserve the high-level principles of the original story and these should correspond to Lehnerts complex plot units. As we build a corpus of stories to train our plot units system with we used the notion of story *interpretations* as another useful abstraction that allows our system to work with different views of a single story, these result in unique subsets of events from an original story. We focus on learning and reasoning on the level of interpretations but unabridged stories are also supported.

Since the primary work in this paper is the identification of representations that can be used for learning to summarize plots we are interested in the class of machine learning frameworks dealing with probabilistic graphical models. These frameworks can model sequential domains in an accessible and tractable way, important for achieving useful results in a short amount of time. Concerning plot units, they have many interesting properties: temporality, event relatedness, and interpretability. These are properties that can be easily exploited by probabilistic graphical models such as Hidden Markov Models, Markov Random Fields, and Conditional Random Fields (see (Rabiner 1989) for a discussion of these in the context of speech recognition). These modeling approaches are suited best to work with sequential domains where events that occur might be dependent on some

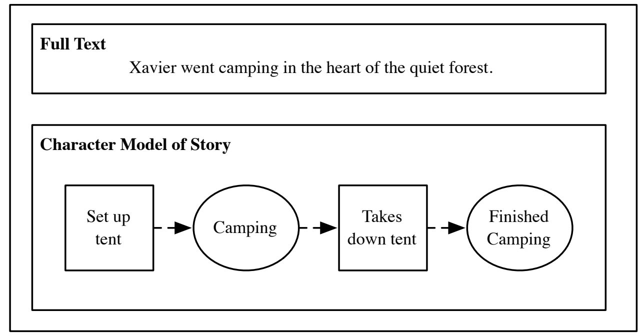

Figure 2: An event node for "Xavier went camping" containing a full-text description of an event and a character model.

of the events that happened previously.

In particular, Conditional Random Fields (CRF) (Lafferty, McCallum, and Pereira 2001; Sutton and McCallum 2006) consider the problem of being given observations X, applying the best labels Y that maximize the conditional probability of the labels. The CRF employs feature functions, which allow us to model useful dependencies in our plot unit models. Unlike the generative Hidden Markov Models, the discriminative CRF allow us to model long range dependencies between events using multiple interacting features and does not require us to explicitly model the distributions that dependencies come from.

Representation of Narrative Input

We represent plot units using a graphical structure where vertices represent story events. There are two classes of links, capturing temporal order (visually represented as dotted edges), and relations between events, e.g., motivation, actualization, termination, and equivalence (visually represented as labeled solid edges).

Events are represented as data structures that neatly organize the progression of action and state changes a protagonist experiences in a story. The inner structure of the event data structure consists of (a) a simplified English sentence for the story event and (b) a provisional character model. The character model captures the causal chain the reader is expecting from their current knowledge of the story. That is, the character model is the prototypical sequence of actions and events that the character would reasonably expect to have happen. For example, in a story called "Xavier goes camping" (Tapiero, den Broek, and Quintana 2002) we would expect a possible event node to look like that in Figure 2.

Modeling Plot Summarization with the CRF

Learning to summarize plots involves learning a model of plot units from previously human-labeled interpretations of stories. Our approach using Lehnerts formulation requires a bottom up parsing process involving the following sequence of subtasks:

1. Label affective state of story events
2. Label relational links between events

3. Label complex plot units as aggregations of primitive plot units and cross-cutting relational links

The three subtasks form a pipeline in which the system successively builds up the knowledge required for the labeling of complex plot units. Affective state labeling of story events is needed to determine relational links, and relational link labels are required for the identification of complex plot units.

Affective State Labeling The first summarization problem, affective state labeling as a sequential labeling problem can be handled by the linear-chain CRF. Without solving the common-sense reasoning problem we are left with having to work under some constraints to make learning for our CRF more tractable. These constraints are the features of the event nodes described earlier. The simplified English feature gives us keywords to compare to supplied and learned dictionaries of words. The character models give us a way to determine how actual events impact the characters perceived plans and goals.

Feature Functions of the Affective State Labeling CRF each have a vote on the likelihood of an event having a positive, neutral, negative, or a no-action neutral label. These describe how the event node features above are used:

- A function that takes in a character model from the previous event and the current event to determine whether an event has impeded or facilitated the characters goal by lengthening or shortening the characters anticipated prototypical sequence.
- A function that recognizes keywords commonly associated with each of the different affective state labels by use of supplied and learned dictionaries.

The first feature function is loosely based on a theory of emotion called cognitive appraisal (Lazarus 1991) in which appraisal variables such as relevance, desirability, and attribution impact ones affective interpretation of events. In this case, we simply determine that if an event increases the number of steps for a character to achieve a goal, it is undesirable, and if an event decreases the number of steps for a character to achieve a goal, it is desirable. Since we only need a negative or positive affective assessment for our CRF, we ignore attribution, which would determine the way in which the negative or positive affects were expressed (e.g., anger/sadness).

Relational Link Labeling There are two problems we face with labeling relational links: relevance and agency. First, to determine whether two events (or an event and a mental state) can have a relational link between them, one must first determine if they are relevant to each other. For example, is wanting a new car related to ones buying of a new car? If so, then our model must consider the possibility that buying a new car actualizes the mental state of wanting a car. In determining relevance, we first consider causal contiguity. That is, instead of looking at the temporal order of events, we look at the causal necessity of events. An event is causally necessary to a following event when the exclusion of the former would mean the later could not then occur (Trabasso, Secco, and van den Broek 1984). Using this notion allows us to break the story into sub-plots so that causally unnecessary events do not interfere with our detection of relevance between events and mental states. Second, we also consider agency of events. Specifically, we determine whether an event is enacted by the protagonist (internal agency) or another character (external agency). For example, "the protagonist buys a car" is internal agency, while "the protagonist is given a car by the insurance company" is external agency.

Features functions to do relational link labeling must be able to discriminate between the four different types of relational links identified by Lehnert. Our feature functions are as follows:

- Termination links: based on dictionary of word dualities.
- Motivation links: based on character model of event nodes.
- Actualization links: based on a dictionary to confirm relevant context between events and analysis of protagonist agency.
- Equivalence links: based on the recurrence of expectations in the character model.

Complex Event Labeling The final task is to label the complex plot units that occur within in interpretation. After having labeled the relational links we now have some idea of the primitive plot units in the story interpretation. However it does not necessarily follow that complex plot units occur just because the correct primitive plot units and links are present; primitive plot units are necessary but not sufficient. This is especially true when relational links exist outside of any primitive plot units but are required for complex plot units. Referring back to Figure 1, (d) and (e) demonstrate the subtle differences between two different complex plot units from Lehnerts work and upon recognizing the existence or non-existence of a relational link in a prospective complex unit we are able to distinguish between plots units that are only slightly different in detail. Note that we only search for complex plot units among causally adjacent events; plot units cannot jump causal chains. Once a complex plot unit is identified, it is "collapsed," meaning we remove it from further consideration. This allows us to recurse and look for complex plot units that may have transcended the previous plot unit; events that were distant may now be contiguous.

Example Story: "Xavier Goes Camping"

This example looks at a story from (Tapiero, den Broek, and Quintana 2002) in which the main character, Xavier, goes camping, experiences his tent being torn, attempts to repair the tent, and finally succeeds in repairing the tent. Figure 3 shows the events from the story with arrows indicating the temporal sequence. There are two interpretations in the figure, each containing a complex plot unit that we have identified for illustrative purposes, although other interpretations and other complex plot units could also exist. In Interpretation A the identified plot units (in order) are as follows:

- *Success*: Xavier like to camp (M) $\xrightarrow{a}$ Xavier went camping (+)

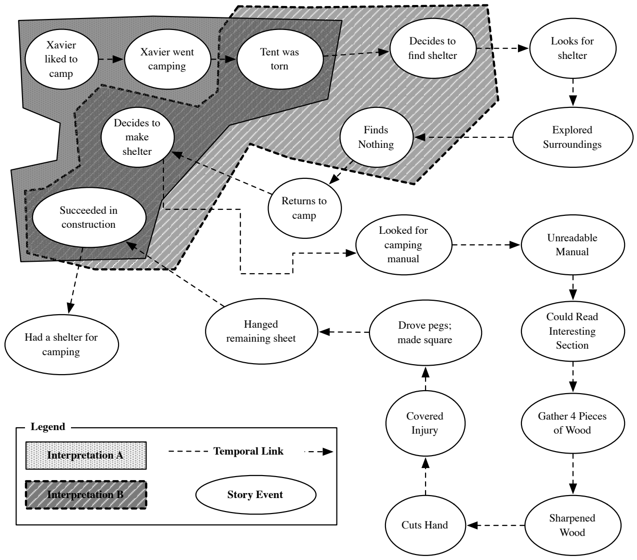

Figure 3: A story graph for "Xavier goes camping" with two interpretations: A (*Success* + *Intentional Problem Resolution*) and B (*Persevering Problem Resolution*).

- *Intentional problem resolution*:
  - *Problem*: Tent was torn (-) $\xrightarrow{m}$ Xavier decides to make shelter (M)
  - *Success*: Xavier decides to make shelter (M) $\xrightarrow{a}$ Xavier succeeded in construction (+)
  - *Resolution*: Xavier succeeded in construction (+) $\xrightarrow{t}$ Tent was torn (-)

The termination (t) link going from "Succeeded in construction" to "Tent was torn" indicates that the successful construction of a shelter extinguishes the issues associated with the tearing of the tent in the earlier event. A similar analysis can be constructed for Interpretation B as well:

- *Intentional problem resolution* (same as that in interpretation A)
- *Failure*: Xavier decides to find shelter (M) $\xrightarrow{a}$ Xavier finds nothing (-)

Persevering problem resolution was not part of Lehnert's original enumeration of complex plot units; there is no indication that Lehnert was attempting to be complete and the *persevering problem resolution* plot unit is a natural extension of the existing set. Note that the *failure* must be nested within the *intentional problem resolution*. The *persevering problem resolution* is a good example of "collapsing": The *failure* primitive plot unit breaks up the encompassing plot-unit such that *intentional problem resolution* complex plot unit is not made of contiguous events. It is not detected by our algorithm until the *failure* plot unit is collapsed.

Future Work and Conclusions

The work in this paper is preliminary. At the time of writing, we have not yet developed a training corpus that can be used to learn a model of plot summarization. However, at the level of analysis provided by Lehnert, we are encouraged by the degree of regularity we can identify in simple, single protagonist stories. Plot summarization is a form of "folk psychological" story understanding– plot units are not attributes of stories themselves, but are commonly understood linguistic terms that can be applied to very distinct stories as a way of package up complicated narrative structures for quick and easy dissemination and digestion. This basic ability to summarize plots can be useful in the development of sophisticated algorithms for reasoning about and evaluating plots in the future.

References

Bruner, J. 1990. *Acts of Meaning*. Cambridge: Harvard University Press.

Cullingford, R. 1981. Sam and micro sam. In Schank, R., and Riesbeck, C., eds., *Inside Computer Understanding*. Hillsdale, NJ: Erlbaum.

Lafferty, J.; McCallum, A.; and Pereira, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*.

Lazarus, R. 1991. *Emotion and Adaptation*. Oxford University Press.

Lehnert, W.; Dyer, M.; Johnson, P.; Yang, C.; and Harley, S. 1983. Boris – an experiment in in-depth understand of narratives. *Artificial Intelligence* 20.

Lehnert, W. 1982. Plot units: A narrative summarization strategy. In Lehnert, W., and Ringle, M., eds., *Strategies for Natural Language Processing*. Hillsdale: Lawrence Erlbaum Associates.

Mueller, E. T. 2004. Understanding script-based stories using commonsense reasoning. *Cognitive Systems Research* 5(4).

Rabiner, L. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2).

Ram, A. 1994. Aqua: Questions that drive the explanation process. In Schank, R.; Kass, A.; and Riesbeck, C., eds., *Inside Case-Based Explanation*. Hillsdale, NJ: Erlbaum.

Sutton, C., and McCallum, A. 2006. An introduction to conditional random fields for relational learning. In Getoor, L., and Taskar, B., eds., *Introduction to Statistical Relational Learning*. Cambridge: MIT Press.

Tapiero, I.; den Broek, P. V.; and Quintana, M.-P. 2002. The mental representation of narrative texts as networks: The role of necessity and sufficiency in the detection of different types of causal relations. *Discourse Processes* 34(3).

Trabasso, T.; Secco, T.; and van den Broek, P. 1984. Causal cohesion and story coherence. In Mandl, H.; Stein, N.; and Trabasso, T., eds., *Learning and Comprehension in Text*. Lawrence Erlbaum Associates.

Wilensky, R. 1981. Pam and micro pam. In Schank, R., and Riesbeck, C., eds., *Inside Computer Understanding*. Hillsdale, NJ: Erlbaum.