Learning Narrative Morphologies from Annotated Folktales

Mark A. Finlayson
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
32 Vassar St., Cambridge, MA 02139 USA
markaf@mit.edu

ABSTRACT

I describe a research program designed to demonstrate the learning of Proppian morphological functions by computer and test if people are sensitive to the presence of those functions in their cultural narratives. The program has two technical components and three stages. The first component is an annotation tool, the Story Workbench, that allows semi-automatic annotation of natural language text semantics by a lightly-trained annotator; the second component is a pattern-extraction algorithm, Analogical Story Merging. In the first stage, I have annotated 16 of Propp's single-move tales translated into English (21,182 words) for their semantics. In the second stage, in progress, I have performed several proof-of-concept demonstrations of the extraction algorithm, and will soon attempt to extract from the annotated tales actual Proppian morphological functions. I detail three metrics I will use to determine success or failure of this extraction. The final stage, yet to begin, is a recall experiment using at least two cultures to test cultural participant's sensitivity to Proppian functions identified by the technique.

1. INTRODUCTION

The morphological functions introduced by Propp [8] remain a tantalizing window into the cultural information embedded in stories. I describe a research program, the first two-thirds of which is covered by my nearly-completed doctoral dissertation, that is designed to (1) demonstrate the learning, by computer, of Proppian morphological functions from actual folktales, and (2) test via cognitive psychology experiment whether cultural participants are sensitive to Proppian functions identified in the folktales of their culture.

First International AMICUS Workshop, October 21, 2010, Vienna, Austria.

This research program has three stages. First (§2) the semantics of a set of folktales must be represented in a computer-understandable manner. Because natural language processing (NLP) is not yet equal to this task, I have developed a computer application, called the Story Workbench, that allows a lightly-trained annotator to annotate free text for its semantics, while allowing the computer to assist where it can. In this stage, I have annotated 16 of Propp's single-move tales, 21,182 words in English, for 17 different meaning representations. Second (§3), an algorithm is needed to extract the Proppian functions from the annotated folktales. I have developed an algorithm called Analogical Story Merging that has shown promise in extracting Proppian functions. There are a number of possible metrics for measuring the accuracy of the results; I detail three (§3.1). Finally (§4), the validity of the extracted functions must be confirmed by experiments on people. I describe an experimental paradigm in which I will test the sensitivity of cultural participants to Proppian functions automatically extracted from their culture's folktales.

2. SEMANTIC ANNOTATION

To automatically extract Proppian functions from text, we need some way of translating natural language text into computer-understandable representations of meaning. Unfortunately, fully automatic NLP is still far from equal to this task; we must therefore resort to manual (or, at best, semi-automatic) semantic annotation. I have developed an annotation tool called the Story Workbench [4] that facilitates semantic annotation by providing a uniform, extensible, user-friendly platform for semantic annotation. Existing NLP techniques may be integrated into the tool, allowing those techniques to contribute automatically-generated annotations where they are able.

The Story Workbench is a fully functional tool, having been used by 12 different annotators so far to annotate various aspects of the semantics of various texts – for example, we recently released a corpus of 24,422 words annotated for referring expressions [6]. The Story Workbench currently has 17 implemented representations, the conjunction of which gives fairly reasonable cover of the basic meaning of a narrative. These representations are:

1. Tokens - location of each word token
2. Multi-word Expressions - words that are made up multiple tokens
3. Sentences - location of each sentence
4. Part of Speech Tags - a Penn Treebank tag for each word token and multi-word expression
5. Lemmas - a lemma (i.e., stem, root form) for each word or multi-word expression not already lemmatized
6. Word Senses - a Wordnet sense for each token or multi-word expression
7. Context-Free Grammar Parse - a CFG parse of each sentence
8. Referring Expressions - locations of all expressions that refer to something
9. Referent Attributes - properties (unchanging attributes) of referents referred to in the text
10. Co-reference Relationships - which referring expressions refer to the same referent (co-refer)
11. Time Expressions - location, type, and value of temporal expressions, as defined by TimeML [9]
12. Events - location, features, and type of event mentions, as defined by TimeML
13. Temporal Relationships - event-event, event-time, or time-time temporal relationships, as defined by TimeML
14. Referent Relationships - event-event, event-referent, or referent-referent non-temporal relationships
15. Semantic Roles - predicate features and arguments, as defined in PropBank
16. Mental State - mental state valencies as consequences of actions, as described by Lehnert [7]
17. Proppian Functions - locations of functions as identified by Propp's monograph

Ten trained annotators have annotated 16 of Propp's single move folktales translated into English, a total of 21,182 words. All 17 of the implemented representations have been double-annotated and adjudicated into a gold-standard for each tale. These particular sixteen tales were chosen for the following reasons. First, Propp identified only 46 of the tales he analyzed. Second, I was able to identify extant translations into English for only 31 of Propp's identified tales, even with the help of Russian speakers searching large numbers of translated collections. Third, of those 31 tales, only 16 were single-move. I targeted single move tales because having only one move in a tale simplifies the observed order of Proppian functions; I hypothesized that this would ease learning the functions, and so should form the first attempt. Thus these 16 single-move, English translations of Propp's original tales comprise the initial set to be analyzed.

The first 16 annotations in the list above will form the raw data for the function extraction algorithm. The final representation, Proppian functions, will be used in the second evaluation metric, namely, comparing my extracted functions with Propp's original analysis.

3. LEARNING MORPHOLOGIES

I have developed an algorithm called Analogical Story Merging (ASM) [5] to extract Proppian functions from the annotated folktales. ASM is a variation of the machine learning technique of Bayesian Model Merging [12]. The algorithm begins by constructing an initial model that explicitly encodes each story as one possible output. I do this by first extracting from each the annotation's of each story a sequence of events, shown as D in Figure 1. Each story's event sequence is then incorporated into the initial model, marked as M_0 in the figure, as a single, linear branch of model states. While there are numerous possible orderings, one of the simplest is make the order of states in the model the same as the order in which their associated events occur in the narration of the story (as opposed the order of events in the *story world*).

ASM then searches the space of *state merges*, where two states, each representing an event, are merged into one. To accomplish this, I define both a merge operation over states, and a *prior* probability function to be used when calculating, via Bayes' rule, the posterior probability of the model given the data. The merge operation takes two states and replaces them by a single state, where the merged state inherits the weighted sum of the transitions and emissions of its parents. Because each state in the initial model represents an event in the story, each merged state represents set of all the events of its parents.

The prior is defined such that smaller models are attributed greater probability than larger models, and models that contain merged states representing sets of similar events are given higher probability than otherwise. In ASM the primary calculation of similarity is done via an analogical mapper, an augmented version of the the Structure Mapping Engine [3]. This mapper assesses the similarity between events, taking into account aspects of those event such as their structure (do the number of arguments match?), their classification (is it a *run* or a *love*?), the identities of other events to which the events in question are connected casually or temporally, the consistency of role assignments (is character A in story 1 consistently mapped to character B in story 2?).

The search space for ASM is quite large, being equal in size to Bell's number, B_n, where n is the number of initial states in the model. Bell's number counts the number of unique partitions of a set of n objects [10], and has been shown [2] to be relatively closely bounded above by equation 1.

$$B_n < \left(\frac{0.792n}{ln(n+1)}\right)^n \tag{1}$$

Because the search space is so large, ASM cannot be expected to do an exhaustive search of the state merge space for a set of real stories. Greedy search is required, with efficient pruning of the search space to ensure that the algorithm converges. I have shown that this approach is feasible in two experiments. The first experiment was reported in [5], and was the first proof-of-concept test of the algorithm using summaries of Shakespearian plays. The initial

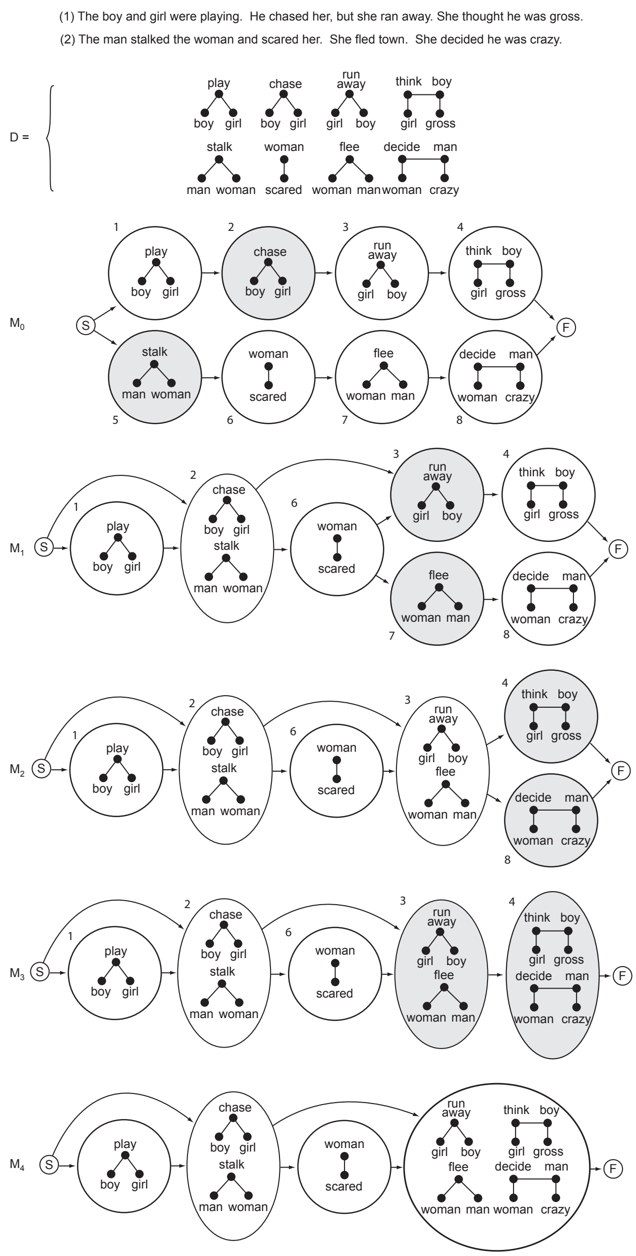

Figure 1: Analogical Story Merging in action. The two stories being merged are written at the top, in (1) and (2). The Story Workbench annotation step produces data structures representing the surface meaning of the story, marked here as D. Each event in each story is then encapsulated in a single state, labeled 1 through 8, in the initial model M_0. ASM searches the space of state merges to find a path to the most probable model, here labeled M_4. From one model to the next, the two states that shaded in the first model are merged together in the second.

model had 48 events across five plays (*Macbeth*, *Hamlet*, *Julius Caesar*, *Othello* and *Taming of the Shrew*) and the search space was pruned by not allowing merges between dissimilar events, but not otherwise optimizing the search. The algorithm converged, and discovered plot similarities that one would expect a human to extract after careful consideration. First, it merged large portions of *Macbeth* and *Hamlet*, the two most similar plays in the set. Second, it merged the ending concluding suicides of *Julius Caesar* and *Othello*, but did not merge these with the (markedly different) suicides of Lady Macbeth and Queen Gertrude. Third, it did not merge the *Taming of the Shrew*, the only comedy in the set, with any of other four tragedies. Numerous other interesting observations may be made, but suffice to say that the algorithm converged on this data and found reasonable patterns.

A second, more recent, experiment has demonstrated that ASM can converge on more complex data. In this experiment, we used four summaries of international conflict situations, written in natural English. These stories were written to illustrate rudimentary plot unit elements (à la Lehnert [7]), in particular, *Revenge* and *Pyrrhic Victory*. After annotation in the Story Workbench, and augmentation of the story graphs with some light commonsense knowledge, each story contained between 34 and 73 states, for a total of 210 states in the initial ASM model. Using a beam search strategy and applying the constraint that all merges in a model must preserve actor mappings across the story, ASM converged and the final graph could be processed to extract the two embedded plot units.

It remains to be seen whether the algorithm, when presented with annotations of real folktales, will be able to extract meaningful functions. Because the extremely large search space induced by 16 folktales of up to 1,800 words each (each folktale potentially containing hundreds of events), I am in the process of augmenting the original ASM implementation to perform efficient, greedy, parallelized beam search, with multiple constraints on valid models, using the 400-node computing cluster available at the MIT Computer Science and Artificial Intelligence Laboratory.

3.1 Evaluation Metrics

I will use at least three metrics to evaluate the output of ASM. The first will be to test the ability of the algorithm to recover patterns purposefully embedded in synthetic data. I will create a synthetic (i.e., artificial) morphology and use it to generate annotations for input into ASM. I will likely start with Propp's own observed morphology over the set of 16 tales that I am analyzing - i.e., including in the morphology only those functions that appear in those 16 tales, and only in those orders. Using this as a skeleton, I will write a generator that outputs, for each Proppian function, a synthetic set of events of the correct semantic character for that function. A set of of synthetic annotations will be generated by this technique, and then fed back into ASM. The functions then discovered by ASM will then be compared with the original synthetic morphology. The measure of success will be an f-measure-like score. The efficiency and reliability of ASM can be evaluated by varying the complexity of the morphology, the number of generated annotations, and the values of the constants in the ASM evaluation functions.

The second metric, perhaps the most interesting, will be to compare with Propp's own analysis the functions that are extracted by ASM when run over the 16 annotated folktales. As we have Propp's original list of functions for these tales, and I will take his analyses as a "gold standard", as it were, to measure the accuracy of the ASM-extracted functions. Beyond the numerical comparison this metric affords, comparing the ASM output with Propp's functions should produce a number of interesting insights. For example, I expect that the annotations I am collecting will not be sufficient to reproduce some of Propp's functions, on account of the wide variation in his level of abstraction. Where ASM breaks down in this case will point to where the abstraction strategy will need to be expanded.

The third metric will be to perform a cross-validation analysis of the set of tales, in which the algorithm is used on different subsets of the 16 tales and the results are compared between the subsets. Such an approach is standard in machine learning studies, and allows testing the sensitivity of the algorithm to variation of input.

4. HUMAN EXPERIMENTS

The true test of this work is whether cultural participants are sensitive to the functions extracted from their own culture's folktales. While there are numerous possible experimental paradigms, in this design we select at least two cultures for study. We will annotate a number of folktales from each culture and extract Proppian functions for each. Using these functions, we will then construct a set of stimuli folktales that are made up primarily of functions from one culture, with the exception of a single function from the other culture. Subjects would then be asked to read these stories and retell them, possibly after a distractor task or delay. Examination of the retold tales should show how subjects treat foreign functions relative to functions from their own culture. If participants preferentially forget or distort foreign functions, we will have fairly clear evidence that people actually detect and extract (and, therefore, probably use) these Proppian functions at some level. There are several possible measures for examining this effect, including reaction time measurements, yes-no judgments of inclusion in the original stimuli (both found in [11]), free-response recall, coded by judges (e.g., [13]), either for a single recall session, or over multiple retellings (such as in a classic study in this area, [1]).

5. ACKNOWLEDGMENTS

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under contract number FA8750-10-1-0076. Thanks to Patrick Winston, Whitman Richards, Ted Gibson, Peter Szolovits, and Josh Tenenbaum for valuable discussion and advice. Thanks to Brett van Zuiden for significant coding and implementation. Thanks to Brett and Nidhi Kulkarni for proof reading.

6. REFERENCES

[1] F. C. Bartlett. Some experiments on the reproduction of folk-stories. *Folklore*, 31(1):30–47, 1920.

[2] D. Berend and T. Tassa. Improved bounds on bell numbers and on moments of sums of random variables. *Probability and Mathematical Statistics*, 30(2), 2010.

[3] B. Falkenhainer, K. D. Forbus, and D. Gentner. The structure-mapping engine. In *Fifth Meeting of the American Association for Artificial Intelligence*, pages 272–277, 1986.

[4] M. A. Finlayson. Collecting semantics in the wild: The story workbench. In J. Beal, P. Bello, N. Cassimatis, M. Coen, and P. H. Winston, editors, *AAAI Fall Symposium on Naturally Inspired Artificial Intelligence*, pages 46–53. AAAI Press.

[5] M. A. Finlayson. *Deriving Narrative Morphologies via Analogical Story Merging*, pages 127–136. New Bulgarian University Press, Sofia, 2009.

[6] R. Hervas and M. A. Finlayson. The prevalence of descriptive referring expressions in news and narrative. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 49–54, 2010.

[7] W. Lehnert. Plot units and narrative summarization. *Cognitive Science*, 4:293–331, 1981.

[8] V. Propp. *Morphology of the Folktale*. Publications of the American Folklore Society, Inc., Bibliographical & Special Series. University of Texas Press, Austin, TX, second edition, 1968.

[9] J. Pustejovsky, J. Castano, R. Ingria, R. Sauri, R. Gaizauskas, A. Setzer, and G. Katz. TimeML: Robust specification of event and temporal expressions in text. In *Proceedings of IWCS-5, the Fifth International Workshop on Computational Semantics*, 2003.

[10] G.-C. Rota. The number of partitions of a set. *The American Mathematical Monthly*, 71(5):498–504, 1964.

[11] C. M. Seifert, R. P. Abelson, G. McKoon, and R. Ratcliff. Memory connections between thematically similar episodes. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 12(2):220–231, 1986.

[12] A. Stolcke and S. Omohundro. *Inducing probabilistic grammars by Bayesian model merging*, volume 862 of *Lecture Notes in Computer Science*, pages 106–118. Springer, Berlin, 1994.

[13] P. van den Broek, E. P. Lorch, and R. Thurlow. Children's and adults' memory for television stories: The role of causal factors, story-grammar categories, and hierarchical level. *Child Development*, 67(6):3010–3028, 1996.