Intelligent Narrative Technologies: Papers from the 2013 AIIDE Workshop (WS-13-21)

Toward Character Role Assignment for Natural Language Stories

Josep Valls-Vargas[1], Santiago Ontañón[1] and Jichen Zhu[2]
[1]Computer Science, [2]Digital Media
Drexel University
Philadelphia, PA 19104, USA
josep.vallsvargas@drexel.edu, santi@cs.drexel.edu, jichen.zhu@drexel.edu

Abstract

In this paper we propose a method for automatically assigning narrative roles to characters in stories. To achieve this goal our proposal is to combine natural language processing techniques with domain knowledge extracted from Propp's morphology of the folktale.

Introduction

In this paper, we present our first step towards automatically extracting narrative structures, such as characters roles and plot from stories written in natural language by bridging computational narrative with existing work on natural language processing (NLP). NLP has seen significant advancements in the recent years with automatic text annotation tools for syntactic and linguistic information. However, with a few exceptions, there has not been much work on automatically extracting high-level semantic and narratological structures. Some early work on story understanding, such as that of Roger Schank's group (Schank and Wilensky 1977; Cox and Ram 1992), focused on the high level constructs humans use to understand, memorize and reason about stories (but they did not focus on NLP). There have been several previous attempts to extract narrative structures from natural language. Finlayson (2008) created the *Story Workbench*, a semi-automatic tool to assist the manual annotation of narrative elements in a text. Similar work has been done by Elson (2012) in *Scheherazade*, focusing on using machine learning and other AI algorithms to help in narrative structure elicitation from the annotated text.

Built on previous work on character identification, we present our preliminary work on automatically assigning narrative roles to characters extracted from text. Some work exists addressing related goals. For example, Calix et al. (2013) presented a machine learning approach to automatically extracting characters directly from natural language, either audio, or text. Goyal et al. (2010) worked on automatically extracting narrative structure represented as characters and their affect states directly from text. Our approach is to combine readily available NLP techniques with domain knowledge.

Character Role Assignment

Automatically extracting knowledge from text is a hard open problem. Different from general purpose text, narrative text follows conventions such as genres and exhibit recurring patterns. For example, Russian folktales generally have *heroes* and *villains*, and the actions these *villains* make are often malicious. At the structural level, there is certain regularity in terms of how different narrative events in these folktales are organized. We will demonstrate that it is possible to use such narratological knowledge to increase the reliability of narrative information extraction. In this section, we present an approach that combines NLP with narrative domain knowledge in order to identify characters from narrative text and their roles (e.g., hero, villain, victim) in the story. For our experiments, we use Russian folktales and Propp's structuralist analysis of them (1973) as our corpus and narrative domain knowledge. Although our current experiments are based on folklores analyzed by Propp, the approach can be generalized to other narrative domains. Figure 1 shows an overview of the role assignment process presented in this paper, composed of two main stages (NLP and Character Role Assignment), which we describe in the following subsections.

Natural Language Processing

Our algorithm may use readily available NLP tools (we used the Stanford CoreNLP) for initially parsing the input natural-language stories. With the purpose of detecting characters in stories, and identifying their narrative roles, we are interested in two main elements of the NLP output: verbs and entities. For each verb, we extract the subject and object as the related entities, generating a set of triplets: $\mathcal{T} = \{t_1, ..., t_w\}$, where each triplet is of the form $\langle$verb, subject, object$\rangle$ (subject or object may be empty). Next, we identify characters in the set of entities by selecting those that appear at least once as subject. The result is a set of characters $\{e_1, ..., e_n\}$. Then, the *character-action matrix generation* (Figure 1) process compiles this information into the *character-action matrix*, C. This matrix summarizes the actions that characters perform to one another in the story. C is a $(n+1) \times (n+1)$ matrix, where n is the number of characters identified in the current story. Each cell $C_{i,j}$ contains the set of verb triplets from $\mathcal{T}$ where e_i is the subject and e_j is the object of the sentence. The matrix has an addi-

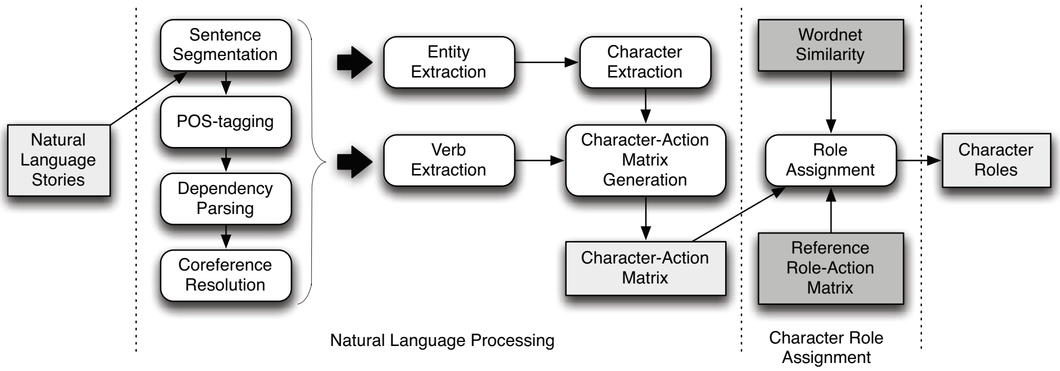

Figure 1: Overview of our approach for identifying characters and their roles via the use of narrative domain knowledge.

Table 1: A sample role-action matrix R in our experiments.

	Hero	Villain	Other	N/A
Hero	talk	fight	rescue, marry	depart
Villain	fight		kidnap, lock	plot
Other	marry			cry
N/A	summon, reward			

tional row and column for those verb triplets that had either no subject or no object.

Character Role Assignment

Propp categorized the narrative role that characters can play in stories into seven broad categories: Hero, Villain, Dispatcher, Donor, (Magical) Helper, Prize, and False hero. We hypothesize that, *given a particular narrative convention, information about how characters behave towards one another can help identify their roles*. To test our hypothesis, we use the domain of Russian folktales and we use Russian folktales and Propp's structuralist analysis of them (1973). Using some of these tales as our corpus and Propp's work as reference, we build a knowledge base of common actions between different roles and use it to identify the narrative roles of the different characters in the story.

Relationships between roles are encoded into a *reference role-action matrix* R (Table 1). As part of the narrative domain knowledge, this matrix is provided as input to our system. Each cell $R_{i,j}$ contains the set of actions that a character (subject) with a role r_i executes over a receiver (object or patient) with a role r_j. The additional *N/A* row and column are for actions performed without a known subject or object.

Character-role identification is performed by 1) assigning one of the m roles to each of the n characters (including an additional *no-role* role for those characters whose role in the story does not match any given role), 2) comparing the assignment with the reference role-action matrix, 3) repeating this process and searching for the assignment that better matches with the reference matrix. We describe these processes in detail below.

Role Assignment. Given n characters and m roles, there are m^n possible assignments of roles to characters. Thus, systematically evaluating all the possible assignments has a prohibitive cost. We use a Genetic algorithm with an initial random population of 80 individuals (role assignments encoded as lists of integers). We perform 1000 iterations of a simple ranked selection with a mutation rate of 2%, and a crossover rate of 90%. We use a simple swap mutator and a single point crossover as recombination operator. In order to speed-up the assignment process we force exactly one character to be assigned the role of the *hero* .

Matching. Each assignment during the search process is matched with the reference role-action matrix R to obtain a *fitness score*. To do this, our algorithm constructs a new character role-action matrix A computed from the character-action matrix C, by replacing the character labels by their assigned roles. Recall that C is a matrix with one row and column per each character, constructed for the current story; A is a matrix with one row and column per each narrative role, constructed from C, and R is a matrix with one row and column per each narrative role, which is given as input to our system, and captures domain knowledge about the typical actions different roles perform towards each other.

A is compared to the reference matrix R cell-by-cell. Each cell in the matrices contains a list of verbs $c_1 = \{v_1, ..., v_r\}$ and $c_2 = \{w_1, ..., w_s\}$. We use the measure proposed by Wu & Palmer (1994; 2004) to calculate the similarity between each pair of verbs $v_i \in c_1$ and $w_j \in c_2$, which we note by $S(v_i, w_j)$. This measure calculates the similarity between two verbs by determining the *least common subsumer* element in the verb taxonomy in WordNet, and then using its depth in the taxonomy to determine the similarity between the two input verbs:

$$S(v_i, w_j) = \frac{2 \times depth(LCS(v_i, w_i))}{depth(v_i) + depth(w_j)}$$

Then, assuming $r \leq s$, we aggregate the values as follows:

$$S(c_1, c_2) = \begin{cases} \frac{\sum_{v_i \in c_1} \max_{w_j \in c_2} S(v_i, w_j)}{s} & \text{if } c_1, c_2 \neq \emptyset \\ 0 & \text{if } c_1 = \emptyset \vee c_2 = \emptyset \end{cases}$$

Intuitively, this measure matches each verb in c_1 with the most similar verb in c_2, and then normalizes by the size of

Figure 2: An excerpt from *The Magic Swan-Geese*, and the corresponding verb-triplets in our structured dataset.

Once upon a time a man and a woman lived with their daughter and small son. $\cdots$ The father and mother went off to work, and the daughter soon enough forgot what they had told her.

$\{\langle live, mother, \bullet\rangle, \quad \langle live, father, \bullet\rangle, \quad \langle live, girl, \bullet\rangle, \langle live, boy, \bullet\rangle\}$
$\{\langle go, father, \bullet\rangle, \quad \langle go, mother, \bullet\rangle, \quad \langle forget, girl, \bullet\rangle, \langle tell, father, girl\rangle, \langle tell, mother, girl\rangle\}$

the largest set of verbs, s. Finally, the values for each cell comparison are added together to obtain a single numeric similarity value.

Selection. The role-assignment and matching processes are repeated and the best assignment is selected.

Dataset

The dataset used in our experiments is composed by the English translation of 8 Russian folktales. In this paper we report experiments performed on a manually annotated version of the dataset. The annotated dataset was built by running the text through an automated sentence boundary detection process and, for each sentence, the characters and verb triplets were annotated manually. Planning an automated NLP extraction, the verbs were annotated using a systematic procedure and no sense disambiguation substitutions were performed. Characters were extracted from the subject and object noun phrases and coreference resolved manually. Figure 2 shows an excerpt of text from one of the folktales in the dataset. The dataset is complemented with a ground truth for role assignments for each of the characters and the roles used in our experiments.

Experiments

We manually crafted 3 versions of the reference role-action matrix R described before.

- R_1: This reference matrix was developed by extracting all the actions described in the 31 Proppian function classes and subclasses. Moreover, we merged the roles of Donor and Helper since, in our dataset, they mostly correspond to the same character. Moreover, the roles of Dispatcher and Prize (and others, such as “victim” or “family member”) are unclear, and thus we grouped them into an “other” role. This resulted in a 7×7 matrix with 506 verbs (6 roles plus one column and row for N/A).
- R_2: This reference matrix was manually created and captures our personal belief of the actions that the different roles might perform upon each other after reading several folktales. This is a 7×7 matrix with 32 verbs.

Table 2: Averaged performance results of our role assignment using the topmost assignment versus weighting all the assignments in the last iteration.

	Top	*Weighted*
R_1	15.10%	48.75%
R_2	31.03%	64.58%
R_3	46.35%	78.99%
Avg.	30.83%	64.11%

- R_3: Finally, we created a simpler matrix, also using our personal belief, with only three roles: Hero, Villain and Other. It was designed to only capture the relationships between these three roles. This is the matrix shown in Table 1.

Table 2 shows the classification accuracy (average percentage of actors with the correct role assigned) obtained using each of the role-action reference matrices R_1, R_2 and R_3. The first column (*Top*), shows the results obtained by selecting the best role assignment found by the Genetic algorithm. The performance is very low (accuracy around 30%). The second column (*Weighted*) uses the role assignments of the entire population in the last iteration weighted by their fitness score and selecting the role with the highest probability. Results improved significantly, reaching classification accuracies of up to 78.99% for reference matrix R_3.

Our results indicate that the matrix similarity metric used for the fitness function used in the Genetic algorithm is effective in determining character roles. However, we observed that, in some situations, the actual ground truth had lower fitness than some of the solutions. This indicates that while useful, either the reference matrices, or the matching procedure being used introduce noise in the results.

Conclusions

In this paper we have presented some preliminary results concerning automatically assigning roles to different characters in a story.

Our system has shown promising results in performing this specific task. We observed that our manually created reference matrices improve the results over the matrix generated from the analysis of Propp’s functions. Using smaller, targeted matrices (like R_3) improves the results and would enable the creation of more efficient single-role classification tools. We plan to increase the size of our dataset with more stories and study the impact of using other reference role-action matrices.

As part of our future work, we will study the use of additional domain knowledge, and the use of feedback loops to reduce the impact of many of the issues we encountered in our experiments. One of our planned first steps is to enhance the extraction and annotation of the verbs to address issues of sense disambiguation, negation and other particles (i.e. phrasal verbs). We also plan on incorporating VerbNet and the use of Levin classes.

References

Calix, R. A.; Javadpour, L.; Khazaeli, M.; and Knapp, G. M. 2013. Automatic Detection of Nominal Entities in Speech for Enriched Content Search. *The Twenty-Sixth* ... 190–195.

Cox, M. T., and Ram, A. 1992. Multistrategy learning with introspective meta-explanations. In *Proceedings of the ninth international workshop on Machine learning*, 123–128. Morgan Kaufmann Publishers Inc.

Elson, D. K. 2012. *Modeling Narrative Discourse*. Ph.D. Dissertation, Columbia University.

Finlayson, M. A. 2008. Collecting semantics in the wild: The story workbench. In *Naturally Inspired Artificial Intelligence, Technical Report FS-08-06, Papers from the AAAI Fall Symposium*, 46–53.

Goyal, A.; Riloff, E.; and Daumé, III, H. 2010. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, 77–86. Stroudsburg, PA, USA: Association for Computational Linguistics.

Pedersen, T.; Patwardhan, S.; and Michelizzi, J. 2004. Wordnet::similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, 38–41. Association for Computational Linguistics.

Propp, V. 1973. *Morphology of the Folktale*. University of Texas Press.

Schank, R. C., and Wilensky, R. 1977. A goal-directed production system for story understanding. *ACM SIGART Bulletin* (63):72–72.

Wu, Z., and Palmer, M. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, 133–138. Association for Computational Linguistics.