Toward Automatic Role Identification in Unannotated Folk Tales

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

Abstract

This paper presents an approach for automatically identifying high-level narrative structure information, particularly character roles, from unannotated folk tales. We introduce a new representation called *action matrices* to encode Propp's narrative theory on character role and their "sphere of action." We tested our approach in a fully automated system (*Voz*) using a corpus of 10 folk tales. Our experimental evaluation shows that action matrices capture useful information for role identification, provides insight into the error introduced by individual steps, and identifies the current bottlenecks.

Introduction

The research area of computational narrative studies how to algorithmically represent, understand and generate stories. Computational narrative systems, especially story generation systems, require the story world to be encoded in structured knowledge representation formalisms (Bringsjord and Ferrucci 1999; Ontañón and Zhu 2011). Currently, knowledge representation of stories is mostly hand-authored, a notoriously time-consuming task requiring expertise in both storytelling and knowledge engineering. One approach to address this well-known "authorial bottleneck" problem is to try to utilize the vast amount of existing written stories in natural language. This will require systems that can process information at both linguistic and narrative levels.

With a few recent exceptions such as (Elson 2012; Finlayson 2008), however, automatically extracting structure-level narrative information such as character roles directly from natural language text has not received enough attention. Such narrative information could be useful for automatically analyzing complex narrative text such as character's social networks (Elson, Dames, and McKeown 2010) or for generating new stories. Our work thus aims to bridge the current gap between computational narrative and Natural Language Processing.

This paper focuses on the problem of *automatic character extraction* and *role identification* from stories in unannotated natural language text. In particular, we use the domain of translated Russian folk tales and Vladimir Propp's Structuralist narrative theory (1973) on character roles developed based on them. Consider the following excerpt:

> *One day, somewhere near Kiev, a dragon appeared, who demanded heavy tribute from the people. He demanded every time to eat a fair maiden: and at last the turn came to the Tsarevna, the princess...*

In Propp's theory, the dragon (*character*) fulfills the specific function of villain (*role*). This structure-level narrative information of roles is important to understand the story as well as its relation to others in its domain. However, as the word "villain" or "hero" rarely appears explicitly in the text, extracting the role information requires combining NLP and narrative theory. A key Proppian insight that we use is that each role has a "*sphere of action*." It defines the core actions of whatever characters fulfilling that role. For example, no matter whether the villain is a dragon or a wizard, its sphere of action centers on villainy, struggle, and pursuit.

This paper presents an approach to role identification in a fully automated system called *Voz*, which uses off-the-shelf NLP packages. Built on our prior work on *semi-automated* role identification from hand-annotated symbolic representations (Valls-Vargas, Ontañón, and Zhu 2013) and on the intermediate step of character identification (Valls-Vargas, Ontañón, and Zhu 2014), this paper evaluates our approach in an end-to-end pipeline from unannotated text and studies how each component's error affect the whole process.

The main contribution of this paper is two fold. First, we developed a symbolic formalism we call *action matrices*. It captures general narratological domain knowledge of character roles and their spheres of action as well as representing story-specific characters and their actions. This matrix-based representation hence allows us to connect the NLP output and narratological knowledge. Second, we designed a *similarity measure* between action matrices. The similarity measure exploits semantic information from knowledge-bases such as WordNet. Overall, we propose a framework to identify Proppian role information by first automatically extracting an action matrix from unannotated text and then comparing it to a reference action matrix which encodes the narratological domain knowledge. We believe that this framework can be generalized to other narrative domains where *characters roles are closely tied to their actions*.

In our experiment with 10 folk tales, our method reached

Table 1: An example *role action matrix* with 3 roles.

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

an average accuracy of 39.65%, significantly higher than the baseline of 25%. It shows that our representation and method captures narratological domain knowledge for role identification. However, the relatively low accuracy rate also shows the limitations in our current approach. For example, our method only focuses on action verbs, while sphere of actions can be expressed in other ways such as action nouns. Another contributing factor is that, when running completely automatically, the preceeding NLP steps introduce error in the system and reduce the overall performance.

In the rest of the paper, we first discuss related work and describe the problem of character role identification. Next, we present our *Voz* system and an empirical evaluation on a corpus of 10 folk tales. Our experimental evaluation provides insight into the error introduced by individual NLP steps and elicit coreference resolution and verb extraction as significant sources of error.

Related Work

Following early work in narrative understanding through Schankian knowledge structures and cognitive processes (Schank and Abelson 1977; Cox and Ram 1992), advances in NLP and related disciplines have led to a renewed interest in extracting and modeling narrative elements from text. Towards this goal, the task of character extraction, related to named entity recognition and nominal actor detection, is a crucial step. Goyal et al.'s AESOP system (2010) explored how to extract characters and their affect states from textual narrative in order to produce plot units (Lehnert 1981) for a subset of Aesop fables. The system uses both domain-specific assumptions (e.g., only two characters per fable) and external knowledge (word lists and hypernym relations in WordNet) in its character identification stage.

Chambers and Jurafsky (2008) proposed using unsupervised induction to learn what they called "narrative event chains" from raw newswire text. In order to learn Schankian script-like information about the narrative world, they use unsupervised learning to detect the event structures as well as the roles of their participants without pre-defined frames, roles, or tagged corpora (2009). Regneri et al (2011) worked on the specific task of identifying matching participants in given scripts in natural language text using semantic (WordNet) and structural similarities in Integer Linear Programming (Wolsey 1998). More recently, Calix et al. (2013) proposed an approach for detecting sentient actors in spoken stories. Based on features in the transcribed textual content using WordNet and speech patterns (e.g., pitch), their system detects sentient actors through supervised learning techniques. Also related is recent research on extracting characters and their social networks from literary fictions (Elson, Dames, and McKeown 2010). Compared to these systems, we take a further step from character extraction to identify the structural roles the characters play.

Recent systems such as *Story Workbench* (Finlayson 2008) and *Scheherazade* (Elson 2012) attempt to solve a similar problem through semi-automatic annotation tools. Although these systems provide valuable data, creating these annotation even with the help of semi-automation can still be time-consuming for human experts.

Automatic Role Identification

A character's role in a given genre defines a typical set of functions performable by and attributes attachable to that character (Prince 2003). Propp categorized characters in Russian folk tales into 7 basic functional roles or character functions (*roles* or *character roles* from now on): Hero, Villain, Dispatcher, Donor, (Magical) Helper, Sought-for-person, and False Hero. Each character role fulfills specific narrative functions and performs its specific "sphere of action." This structural-level regularity, or genre convention in narrative terminology, is instantiated into different characters and specific actions in specific stories. Our goal is to capture this regularity and identify role information regardless of the specificity of how characters and actions are constructed at the story and discourse levels.

Problem Statement: given an unannotated folk tale, the problem we address in this paper is how to extract the set of characters $\{a_1, ..., a_n\}$ in the text, and how to identify which is the most likely narrative role from a given set of roles $\{r_1, ..., r_m\}$ that each character plays.

Knowledge Representation: A *role action matrix* R represents the set of actions that a given set of roles perform upon each other. Table 1 shows a simple role action matrix. For a given set of m roles, a role action matrix R is an $(m+1) \times (m+1)$ matrix where each row and each column represent one of the roles. A given cell $R_{i,j}$ contains the set of actions that characters with role r_i perform upon characters with role r_j. Each action is represented as a verb. The additional N/A row and column are used for those actions that do not have an executor or a receiver, such as intransitive verbs. Also, note the diagonal of the matrix need not be empty since a) characters can perform actions upon themselves (transform) and, b) different characters in the same role can interact with one another (talk).

Similarly, a *character action matrix* C is an action matrix whose rows and columns correspond to specific characters.

Dataset. Our dataset contains 10 Russian folk tales translated into English text. We selected stories studied by Propp, 6 of which were collected by (Malec 2010) and 4 by (Finlayson 2012). Our corpus of the 10 stories include characters with all 7 Proppian roles.

To reduce NLP parsing and coreference issues at the discourse level, we manually removed: 1) dialogues, and 2) passages where the narrator addressed the reader directly (e.g. "If you think of it ..." where "you" refers to the

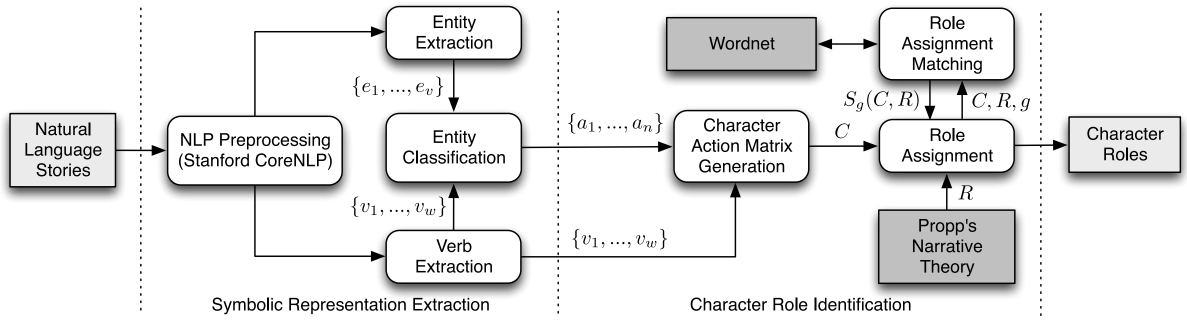

Figure 1: The architecture of *Voz*.

reader). Our edited dataset contains 403 sentences ($\mu = 14.46$ words). The stories range from 19 to 53 sentences ($\mu = 40.30$ sentences, $\sigma = 11.18$).

Although the stories are relatively short, they are nevertheless complex. For example, in our Story S_1, a mythical creature called Morozko gave the hero "a warm fur coat and downy quilts." In order to understand Morozko is *helping* the hero, the context of the forest in the winter is important. Furthermore, some actions need to be inferred. The text only contains how the step-sister of the hero answered Morozko's question rudely. In the next scene, her mother "saw the body of her daughter, frozen by an angry Morozko," leaving out Morozko's direct actions to inference.

Role Identification in the *Voz* System

The *Voz* system contains two main stages: *Symbolic Representation Extraction* (built upon off-the-shelf NLP packages), and *Character Role Identification*, where our main contribution resides (Figure 1). Voz builds upon our previous work on semi-automated role identification from hand-annotated symbolic representations (Valls-Vargas, Ontañón, and Zhu 2013) and character classification (Valls-Vargas, Ontañón, and Zhu 2014). A new contribution here is their integration into an end-to-end pipeline from unannotated text.

Symbolic Representation Extraction

At this stage, *Voz* extracts a list of characters and a list of verb triplets using the following 3 components.

NLP Preprocessing. *Voz* uses the Stanford CoreNLP suite to segment the input text into sentences and compute the following: word POS tags (part-of-speech, e.g. whether a word is a noun, a verb, etc.), syntactic parse trees, typed dependency lists (e.g., relations between adjectives and their associated nouns, verbs and their subject and complements), lemmatization and anaphora and coreference information (e.g., "Shabarsha," "laborer," "he" referring to the same character).

Entity Extraction. *Voz* traverses each sentence's parse tree looking for any noun phrase (NP) node. For each NP node, *Voz* does the following: if the subtree contains nested clauses or phrases, or an enumeration, it is marked as a *compound-entity*; otherwise, the NP node is marked as a potential *entity*, and its subtree is not further explored.

Then, each potential entity for which *Voz* finds a noun, personal pronoun or possessive pronoun in its associated subtree is marked as actual *entity*. All compound entities are populated with all the entities that they contain in their respective subtrees (compound-entities are represented as sets of entities). For example, "father and mother" yields two entities and a compound entity with two children. A future occurrence of "they" can refer to the compound entity. Finally, we use the previously extracted coreference information to determine whether different entities refer to the same character or object. *Voz* generates a set $\{e_1, ..., e_v\}$, where each e_i is a *coreference group*, and is composed of the set of entities that the coreference resolution system deemed as pointing to the same referent.

Verb Extraction. To detect the verbs of a sentence *Voz* uses the typed dependencies. Specifically *Voz* looks at dependencies of the types "nominal subject", "passive nominal subject" and "expletive" where the *head word* token is POS-tagged as a verb (to exclude copula). The dependent of the typed dependency is considered the subject, and the rest of dependencies of each verb are explored to extract the direct object, indirect object and prepositional objects. The first available is used as the object of the verb. Then, for each verb, a triplet of the form $v = \langle$verb lemma, subject, object$\rangle$ is generated, where subject and object are coreference groups. The final output is a set of triplets $\{v_1, ..., v_w\}$.

Entity Classification. This component determines which entities are characters by extracting a feature vector from each entity or coreference group in $\{e_1, ..., e_v\}$. We compiled a list of 193 syntactic and semantic features, 21 of which are based on Calix et. al. (2013). These features include whether there is a personal pronoun, whether the entity is the subject of a verb, whether there are nouns that have certain relationships in WordNet, etc. More details can be found in (Valls-Vargas, Ontañón, and Zhu 2014). From the features, *Voz* determines whether an entity is a character using supervised machine learning. We use a Case-Based Reasoning (CBR)approach where the annotated entities in our dataset as a case base. We defined a continuous version of the Jaccard similarity measure, that uses the Information Gain of each feature (Quinlan 1986) to weight the contribution of each feature to the similarity (Valls-Vargas, Ontañón, and Zhu 2014). The output of this component is the set of characters $\{a_1, ..., a_n\} \subseteq \{e_1, ..., e_v\}$ from the story.

Character Role Identification

Using the sets of characters and verb triplets from the Symbolic Representation Extraction stage, this second stage computes the expected role assignment. It generates and compares the story-specific character action matrix with the domain knowledge of role and their sphere of action, represented as a manually constructed *reference role action matrix* R. We constructed three versions of R by compiling the actions in Propp's functions or our own abstraction of actions of different roles from the stories (details in "Experiments"). Using R as an input to *Voz*, the role identification process is performed as the following.

Character Action Matrix Generation. *Voz* compiles the verb and character information extracted in the previous stage into a *character action matrix*, C. C summarizes all the actions that each character performed to one another in the story. This matrix captures *story-specific* relationships between characters and actions. Similar to the role action matrix, 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 where a_i is the subject and a_j is the object of the sentence.

Role Assignment. Given a set of m possible roles, role assignment is performed by the following iterative process: 1) assigning one of the m roles to each of the n characters, including an additional N/A role for those characters that play no clear role or may be misclassified as characters, 2) then matching the assignment with the reference role action matrix R using the process discussed in the "Role Assignment Matching" section below, 3) repeating this process and selecting the assignment that best matches the reference matrix. Given n characters and m roles, there are m^n possible assignments of roles to characters. Due to limitations in coreference resolution, a large number of characters may be inaccurately detected in a given story. For example, 79 characters were identified in S_5 from our dataset and thus a systematic evaluation of all possible role assignments not viable. We use a greedy search hill-climbing approach with a random initialization. For each assignment a successor is selected by generating all possible one-role variations of the current assignments, and choosing the one that maximizes the similarity matching. We executed this algorithm for 100,000 iterations with random restarts when the search is stuck for more than 1,000 iterations. Each role assignment g maps each character a to a role $g(a)$.

Role Assignment Matching. In order to evaluate a role assignment g, *Voz* matches the character action matrix C against the reference role action matrix R via the assignment g. Each cell in R or C contains a set of verbs. For example, cell $R_{i,j} = \{w_1, ..., w_r\}$, contains the actions that characters of role r_i perform on characters of role r_j. Notice that verbs in the reference matrix are prototypical (e.g. "fight") and will most likely not exactly match the verbs extracted form the text, which correspond to specific actions (e.g. "drowned him"). To address this problem, *Voz* uses the measure $S(v, w)$ proposed by Wu & Palmer (Wu and Palmer 1994; Pedersen, Patwardhan, and Michelizzi 2004), which can assess the similarity of two verbs v and w using WordNet. Specifically, it calculates the *least common subsumer* (LCS) verb in the verb taxonomy in WordNet, and then uses its depth in the taxonomy (i.e. the distance from the LCS to the root of the taxonomy) to determine the similarity between the two input verbs:

$$S(v, w) = \frac{2 \times depth(LCS(v, w))}{depth(v) + depth(w)}$$

To compare the cells $C_{i,j}$ and $R_{k,l}$ of the matrices, we propose the following similarity measure:

$$S(C_{i,j}, R_{k,l}) = \begin{cases} \sum\limits_{v \in C_{i,j}} \max\limits_{w \in R_{k,l}} \frac{S(v,w)}{N} & \text{if } C_{i,j}, R_{k,l} \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

where $N = max(|C_{i,j}|, |R_{k,l}|)$. Intuitively, this measure matches each verb in $C_{i,j}$ with the most similar verb in $R_{k,l}$, and then normalizes by the size of the largest set of verbs.

The similarity measure between the two matrices C and R assuming assignment g is then defined as:

$$S_g(C, R) = \frac{\sum\limits_{i=1}^{n+1} \sum\limits_{j=1}^{n+1} S(C_{i,j}, R_{g(a_i), g(a_j)})}{(n+1)^2}$$

Experiments

We evaluated our approach using a dataset of 10 folk tales, including the performance of coreference resolution, verb extraction, entity classification, and role identification. Even though some components use off-the-shelf packages, evaluating their performance is crucial to understand where errors occur and how they propagate in the complete pipeline.

Coreference Resolution. Coreference is foundational to the subsequent steps in role identification. Table 2 compares the performance of the Coreference Resolution component used by *Voz*, from Standford CoreNLP, to human hand-annotated ground truth. For each story in the dataset, a human annotator identified by hand its number of distinct characters (*# Chars.*), number of mentions to entities (*# Mentions entities*), and the subset of those that were mentions to characters (*chars.*). We then recorded the number of coreference groups found by the Standford CoreNLP, and among them, we manually counted the number of those related to the characters the human annotator identified. Ideally, mentions to the same character, object or location should be gropuped into a single coreference group. We then measured the percentage of accurate coreferences (coreference groups that do not contain any false positive links, and hence only contain mentions to a single referent). As Table 2 shows, the percentage of accurate coreference groups is very high (*% Accurate Coref.*), but when we compare the number of coreference groups referring to characters with the actual number of characters, many mentions to the same character were not merged into the same coreference group as they should have

Table 2: Number of characters and mentions, coreference resolution and verb extraction results for each story in our dataset.

Story	# Chars.	# Mentions entities (chars.)	# Coreference groups (chars.)	% Accurate Coref.	# Verb triplets (GT)	Verb Ext. (P/R/F)
S_1	7	149 (90)	68 (37)	97.30%	54 (51)	0.44 / 0.47 / 0.46
S_2	5	114 (44)	64 (20)	90.00%	36 (33)	0.53 / 0.58 / 0.55
S_3	11	181 (87)	102 (55)	81.82%	66 (47)	0.50 / 0.70 / 0.58
S_4	9	250 (115)	110 (31)	77.42%	74 (50)	0.39 / 0.58 / 0.47
S_5	11	189 (128)	106 (79)	96.20%	63 (55)	0.46 / 0.53 / 0.49
S_6	6	145 (80)	73 (32)	87.50%	48 (47)	0.48 / 0.49 / 0.48
S_7	9	170 (77)	66 (33)	69.70%	46 (46)	0.85 / 0.85 / 0.85
S_8	11	166 (79)	65 (40)	65.00%	47 (48)	0.83 / 0.81 / 0.82
S_9	5	154 (64)	78 (55)	69.09%	48 (51)	0.77 / 0.73 / 0.75
S_{10}	7	217 (93)	67 (30)	76.67%	65 (65)	0.77 / 0.73 / 0.75
	Sum: 81	Sum: 1735 (857)	Sum: 799 (412)	81.80%	Sum: 547 (493)	0.60 / 0.65 / 0.62

been. For example, story S_1 has 7 characters, but CoreNLP identifies 37 different coreference groups corresponding to characters. We evaluated the MUC precision and recall scores (Pradhan et al. 2011), which echo these results: we obtained a relatively high precision, 0.65, but very low recall of 0.03. The overall results are much lower than those reported in the Stanfords Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task (2011).

We believe that this low performance is partially due to the different characteristics of our domain, compared to the newswire and formal or legal text domain that the NLP tools are trained on. For example, in folk tales, a single character can be referred to with different nouns such as "girl," "sister," "maiden." Furthermore, certain anthropomorphic characters such as a talking tree are referred using the pronoun "it" which is usually not used to refer to characters.

Verb Extraction. In order to test our verb extraction component, we compared the extracted verbs from each story against the verbs annotated in our dataset. Across the 10 stories, we observed an average precision of 0.60, average recall of 0.65 and an overall F-measure of 0.62. The right-hand side of Table 2 shows detailed results of the number of verb triplets extracted for each story and the number of verb triplets in the annotated ground truth and the precision, recall and F-measure. *Voz* extracts 547 verb triplets from the output of the parser. 61.85% of the 493 verbs annotated in our triplet ground truth are correctly identified. There are actually more verb triplets extracted than there are in our ground truth, mainly because of parsing problems.

Entity Classification. The process of classifying entities into characters and non-characters is crucial because any false positives will add noise to the character action matrix and directly impact the performance of role assignment. We compared the new classifier employed by *Voz* (*WCJaccard*) against two other methods: 1) a rule based classifier (*Baseline*) the checks wether the entity appears as the subject of a verb or includes a pronoun as used in our previous work (Valls-Vargas, Ontañón, and Zhu 2013) 2) a decision tree (J48) using the full feature vector for each entity (we used the J48 decision tree implementation in Weka (Hall et al. 2009)). We evaluated the performance of entity classification of all three different methods in five different scenarios:

- *Before Coreference*: identifying which of the entities are characters before doing coreference resolution. Entities were classified individually.
- *Automatic Coreference*: using automatic coreference resolution, feature vectors were computed for each coreference group, and the system classified each coreference group.
- *Annotated Coreference*: the same, but doing coreference resolution manually (without errors).
- *Automatic Coreference Vote*: in this scenario, entities are classified first, as in the *Before Coreference* scenario, and then each coreference group is automatically identified as a character or not based on whether the majority of entities it contains are classified as characters or not.
- *Annotated Coreference Vote*: the same, but doing coreference resolution manually.

Table 3 shows the performance of the three classifiers we evaluated for this task. Both the *J48* and *WCJaccard* perform better than the baseline. We noticed the classifiers performed better classifying the individual entities extracted from the stories than the coreference groups, which provided the motivation for experimenting the last two scenarios, using voting to classify coreference groups. Experiments seem to indicate that performance is better with automatic coreference than with annotated coreference, however, we believe that to be an artifact of the fact that automatic coreference groups are smaller (leading to more instances in the case-base). Also, we would like to point out that our annotated coreference only included character coreference groups (all non character entities were not coreferences, which biased the dataset and is responsible for the low precision values in this scenario). For the remainder of the experiments we used a this voting approach using the *WCJaccard* classifier, which provides the best results, as shown on Table 3.

Character Role Identification. We experimented with three different reference role action matrices:

- R_1: created by extracting all the actions described in the 31 Proppian functions and subfunctions (Propp 1973).

Table 3: Character classification performance on individual entities (Accuracy / Precision / Recall).

Classifier	*Before Coref.* (P / R / F)	*Automatic Coref.* (P / R / F)	*Annotated Coref.* (P / R / F)	*Automatic Coref.* Vote (P / R / F)	*Annotated Coref.* Vote (P / R / F)
Baseline	0.59 / 0.89 / 0.71	0.46 / 0.87 / 0.61	0.16 / 0.93 / 0.28	0.46 / 0.88 / 0.61	0.19 / 0.95 / 0.32
J48	0.79 / 0.59 / 0.68	0.67 / 0.57 / 0.61	0.16 / 0.93 / 0.28	0.68 / 0.55 / 0.61	0.18 / 0.73 / 0.29
WCJaccard	0.87 / 0.87 / 0.87	0.80 / 0.79 / 0.80	0.37 / 0.77 / 0.50	0.80 / 0.85 / 0.82	0.52 / 0.82 / 0.63

We merged the roles of Donor and Helper since, in our dataset, they mostly correspond to the same character. This resulted in a 7×7 matrix with 506 verbs.

- R_2: manually created capturing our own abstraction of the actions by characters of different roles. Like R_3, we extracted what we observed as prototypical actions of different roles without using the specific verbs from the text. Although R_1 is more theoretically sound, its large amount of verbs and overlapping roles imposes a practical challenge. As we rely on similarities, the more verbs we include, the more likely the system will find matchings that are not as meaningful. Thus, R_2 and R_3 aim at capturing the most relevant information, reducing the number of verbs. Admittedly, a limitation is that this adds more dependency on human processed data. Moreover, the roles of Dispatcher, Sought-for-person, Victim and Family Member are unclear, and thus we grouped them into an "other" role. This is a 6×6 matrix with 32 verbs.
- R_3: a simpler 4×4 matrix, with only three roles (Hero, Villain and Other) manually designed to capture only the relation between these three roles (Table 1).

Table 4 shows the role identification performance (average percentage of characters with the correct role assigned) obtained using each of the reference role action matrices. *Baseline* corresponds to assigning roles at random (including the N/A role present in each matrix). *Automatic* reports the results running *Voz* fully automatically. In *Filtered*, we provide the system with a ground truth for the coreference and character identification tasks. Still the verb extraction and, more importantly, the action triplets are generated by the system. Finally, *Annotated* reports results when the character action matrix C is actually hand-authored (thus removing error introduced by coreference resolution, character identification and verb extraction).

As Table 4 shows, our system identified roles with twice the performance of the baseline in the annotated scenario (specially for matrices R_2 and R_3). This indicates that the information present in the role action matrices is in fact indicative of the roles characters play in stories, thus, confirming our working hypothesis in this paper. Moreover, fully automated results perform only slightly above the baseline. This lower performance, compared to the annotated scenario, is due to error in the early stages of the NLP processing. We expect that the performance of *Voz* running fully automatically will improve by training the off-the-shelf NLP components with a corpus similar to our application domain, which is part of our future work.

A closer analysis of the results, showed that the performance of the system was very high for certain stories (in some stories more than 80% of the characters had their roles consistently identified correctly), but low for other stories, where characters behave very different from what is typical for their roles. For example, in one of our stories, the Hero barely performs any action and always requests the Helper to perform all the tasks for him. Also, our approach tends to favor roles with a bigger "sphere of action" (i.e. their corresponding cells in the reference role action matrix contain more verbs) and are more likely to find matching verbs in the character matrix.

Table 4: Role identification performance comparison.

	Baseline	*Automatic*	*Filtrered*	*Annotated*
R_1	14.29%	11.54%	15.85%	20.94%
R_2	16.67%	23.76%	21.89%	35.60%
R_3	25.00%	39.65%	41.88%	44.12%
Avg.	18.65%	24.99%	26.54%	33.56%

In summary, our results show that a larger amount of NLP errors introduced into the system are verb triplet extraction and coreference resolution, while character identification can be performed with significant accuracy. Concerning role identification, our results indicate action matrices contain useful information for role identification, but we believe results can be significantly improved by moving beyond action verbs to capture character relations in stories.

Conclusions

We presented an approach to automated role identification from unannotated folk tales. Our representation formalism of action matrices is used to bridge the domain knowledge of Propp's theory and story-specific character actions. Using off-the-shelf NLP tools and the Wordnet knowledge base, our automated *Voz* system extracts characters from the text and identifies their roles using a similarity measure that exploits knowledge encoded as action matrices. In our evaluation on a corpus of 10 stories, experiments show that *action matrices* contain information that can be used for role identification, and we believe that this approach can be generalized to a broader range of story genres and narratological theories. Our evaluation of the pipeline provides many insights into which are the steps that introduce error into the system and the potential for automatically extracting narrative information from text using our *action matrices*.

For our future work, we will study the use of additional sources of common sense knowledge (ConceptNet, VerbNet) to improve the performance of the symbolic representation extraction. We would like to include information about the relations between Propp's functions into the process in order to improve role identification performance.

References

Bringsjord, S., and Ferrucci, D. 1999. *Artificial Intelligence and Literary Creativity: Inside the Mind of BRUTUS, a Storytelling Machine*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc.

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 International FLAIRS Conference* 190–195.

Chambers, N., and Jurafsky, D. 2008. Unsupervised Learning of Narrative Event Chains. *ACL* 94305(June):789–797.

Chambers, N., and Jurafsky, D. 2009. Unsupervised learning of narrative schemas and their participants. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* 2(August):602–610.

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.; Dames, N.; and McKeown, K. R. 2010. Extracting social networks from literary fiction. In *Proceedings of the Fourty-Eighth Annual Meeting of the Association for Computational Linguistics*, 138–147. Association for Computational Linguistics.

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 *Proceedings of the AAAI Fall Symposium: Naturally-Inspired Artificial Intelligence*, 46–53.

Finlayson, M. A. 2012. *Learning narrative structure from annotated folktales*. Ph.D. Dissertation, Massachusetts Institute of Technology.

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*, 77–86. Association for Computational Linguistics.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter* 11(1):10–18.

Lee, H.; Peirsman, Y.; Chang, A.; Chambers, N.; Surdeanu, M.; and Jurafsky, D. 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, 28–34. Association for Computational Linguistics.

Lehnert, W. G. 1981. Plot units and narrative summarization. *Cognitive Science* 5(4):293–331.

Malec, S. 2010. Autopropp: Toward the automatic markup, classification, and annotation of russian magic tales. In *Proceedings of the First International AMICUS Workshop on Automated Motif Discovery in Cultural Heritage and Scientific Communication Texts*, 112–115.

Ontañón, S., and Zhu, J. 2011. On the role of domain knowledge in analogy-based story generation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Two*, 1717–1722. AAAI Press.

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.

Pradhan, S.; Ramshaw, L.; Marcus, M.; Palmer, M.; Weischedel, R.; and Xue, N. 2011. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, 1–27. Association for Computational Linguistics.

Prince, G. 2003. *A dictionary of narratology*. Lincoln: Univ of Nebraska Press.

Propp, V. 1973. *Morphology of the Folktale*, volume 9. University of Texas Press.

Quinlan, J. R. 1986. Induction of decision trees. *Machine learning* 1(1):81–106.

Regneri, M.; Koller, A.; Ruppenhofer, J.; and Pinkal, M. 2011. Learning Script Participants from Unlabeled Data. *RANLP*.

Schank, R. C., and Abelson, R. 1977. Scripts, goals, plans, and understanding.

Valls-Vargas, J.; Ontañón, S.; and Zhu, J. 2013. Toward character role assignment for natural language stories. In *Proceedings of the Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 101–104.

Valls-Vargas, J.; Ontañón, S.; and Zhu, J. 2014. Toward automatic character identification in unannotated narrative text. In *Proceedings of the Seventh Workshop in Intelligent Narrative Technologies (INT)*.

Wolsey, L. A. 1998. *Integer programming*, volume 42. Wiley New York.

Wu, Z., and Palmer, M. 1994. Verbs semantics and lexical selection. In *Proceedings of the Thirty-Second annual meeting on Association for Computational Linguistics*, 133–138. Association for Computational Linguistics.