Inducing Stereotypical Character Roles from Plot Structure

Labiba Jahan
Department of Mathematics
& Computer Science
Augustana College
639 38th Street,
Rock Island, IL 61201 USA
labibajahan@augustana.edu

Rahul Mittal, Mark A. Finlayson
Knight Foundation School of
Computing and Information Sciences
Florida International University
11200 S.W. 8th Street,
Miami, FL 33199 USA
{rmitt008, markaf}@fiu.edu

Abstract

Stereotypical character roles—also known as *archetypes* or *dramatis personae*—play an important function in narratives: they facilitate efficient communication with bundles of default characteristics and associations and ease understanding of those characters' roles in the overall narrative. We present a fully unsupervised k-means clustering approach for learning stereotypical roles given only structural plot information. We demonstrate the technique on Vladimir Propp's structural theory of Russian folktales (captured in the extended ProppLearner corpus, with 46 tales), showing that our approach can induce six out of seven of Propp's *dramatis personae* with F_1 measures of up to 0.70 (0.58 average), with an additional category for minor characters. We have explored various feature sets and variations of a cluster evaluation method. The best-performing feature set comprises plot functions, unigrams, tf-idf weights, and embeddings over coreference chain heads. Roles that are mentioned more often (Hero, Villain), or have clearly distinct plot patterns (Princess) are more strongly differentiated than less frequent or distinct roles (Dispatcher, Helper, Donor). Detailed error analysis suggests that the quality of the coreference chain and plot functions annotations are critical for this task. We provide all our data and code for reproducibility[1].

1 Introduction

Stereotypical characters are characters that both play an important role in the plot of a story and fit into recognizable categories. In general, characters are central to every narrative and drive the action forward, and stereotypical character roles include both common, context-independent roles such as Hero, Villain, or Victim, as well as culturally-specific roles such as the Donor (in, for example, Russian tales) or the Trickster (in, for example, Native American tales). Referred to alternatively as *archetypes* (Abrams and Harpham, 2014) or *dramatis personae* (Propp, 1968), stereotypical character roles are crucial aids to narrative understanding: they facilitate efficient communication with bundles of default characteristics and associations and ease understanding of the purpose of those character in the overall narrative (Robbins, 2005). Beyond demonstrated cognitive effects, stereotypical character roles are useful for NLP tasks such as narrative generation (Gervás, 2013), interactive dialogue generation (Rowe et al., 2008), and sentiment analysis (Bhaskaran and Bhallamudi, 2019).

Prior work has demonstrate the utility of pre-identified roles. But how do we learn the roles in the first place? There have been several approaches to this task, but all prior work incorporated some *a priori* knowledge of the possible stereotypical roles in the model, for example, results of manual qualitative analyses (Harun and Jamaludin, 2016), an archetype ontology (Groza and Corde, 2015), or feature vectors of archetype information (Valls-Vargas et al., 2016). Ideally a solution to this task will learn roles from the data in a completely unsupervised manner. We present just such an approach here, a k-means-based unsupervised clustering using plot functions as the key feature: we show that if you know characters' involvement in plot functions for a corpus, we can automatically induce the stereotypical roles with reasonable performance.

The paper proceeds as follows. To motivate our approach we begin by describing prior work on learning stereotypical character roles (§2). We next describe our corpus (§4), followed by the experimental setup, including cluster assignment methods, features, and clustering models (§5). We present the results (§6) and analyze the error patterns of the system, discussing various aspects, which leads us to a discussion of future work (§7). We conclude with our contributions (§8).

[1] https://doi.org/10.34703/gzx1-9v95/DD6SEN

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 492–497
November 7–11, 2021. ©2021 Association for Computational Linguistics

2 Related Work

Vladimir Propp (1895–1970) was a Russian folklorist who provided one of the first classic accounts of stereotypical character roles in literary theory (Propp, 1968). Propp studied a corpus of 100 Russian Hero folktales, and in his analysis proposed 31 plot functions and seven stereotypical character roles (which he called *dramatis personae*): *Hero*, *Villain*, *Donor*, *Helper*, *Princess*, *Dispatcher*, and *False Hero*. While *Hero* and *Villain* are fairly universal, roles such as *Donor* and *False Hero* are somewhat culturally specific.

There is a limited amount of prior work on learning or using stereotypical character roles in stories. One body of work uses roles, but does not automatically extract them. For example, Valls-Vargas et al. (2014b) built upon their work in character identification (Valls-Vargas et al., 2014a) to assign stereotypical roles to characters. The authors encoded Propp's "sphere of action" (Propp, 1968, §6) into a role action matrix and used a greedy similarity matching approach to assign roles to characters achieving 33.56% accuracy when using manually extracted characters. Similarly, Skowron et al. (2016) designed a system to classify characters in action movies into categories such as *Hero*, *Antagonist*, *Spouses*, and *Sidekicks* using graph and n-grams features, with an overall performance of 0.43 F_1. Groza and Corde (2015) integrated Propp's seven *dramatis personae* into an existing ontology, and then exploited constraints of character roles to reason over the ontology, inferring such things as family relationships and whether an entity was a main character. The model achieves 74% accuracy and outputs major characters who belong to one the seven types, but does not classify them more precisely.

Other work has tackled unsupervised clustering of characters, but either at more abstract levels or not quantitatively evaluated. The level of abstraction is important, because the more abstract a character role, the more likely it is to be found across cultures: unlike automatic character identification (Jahan et al., 2020), which is generalizable across domains, stereotypical character roles depend strongly on the cultural background of the text. For example, Chen et al. (2019) used a minimum span clustering approach to group characters into *core*, *secondary* and *peripheral* categories using a character network; such categories, while useful for stereotypical role learning, are not themselves culturally-specific stereotypical roles. Bamman et al. (2013) identifies the what they call the *persona* of characters—similar to a stereotypical character role—by clustering agent and patient actions as well as the adjectives used to describe the characters. Their model achieves 42% purity at best between the models of the same size. Following a similar persona definition, Bamman et al. (2014) developed the *BookNLP* pipeline to extract narrative information from English novels. The model is hierarchical and assigns multiple personas to a characters, and the authors used the analysis to explore the relationship between character persona and author style and literary effects; however, the reliability or performance of the actual persona extraction was not quantitatively evaluated.

Stereotypical roles are also useful in other NLP tasks. Gervás (2013) explores the use of Propp's 31 plot functions and seven *dramatis personae* to generate stories, while Rowe et al. (2008) propose a model to generate role-appropriate dialogues for different character archetypes in an interactive environment. Another recent work (Bhaskaran and Bhallamudi, 2019) looks at stereotypical gender and occupational roles to identify bias in sentiment analysis models.

3 Propp's Morphology

Vladimir Propp (1895-1970) was a Russian folklorist who wrote one of the first classic analyses of stereotyped character roles in literary theory (Propp, 1968). Propp analyzed 100 Russian folktales and introduced seven stereotypical character roles, listed below, which were connected to 31 basic structural elements or plot functions typical of the Russian hero tales he analyzed, as shown in Table 1.

Hero The role model of a story.

Villain The negative character who creates struggles for the hero.

Donor The character who provides some magical object to the hero.

Helper The character who helps the hero.

Princess The character who becomes a companion of the hero.

Dispatcher The character who illustrates the need for the hero's quest and sends the hero off.

False Hero The character who takes credit for the hero's actions.

Symbol	Name	Description
β	Absentation	One of the members of a family absents himself from home.
γ	Interdiction	An interdiction is addressed to the hero.
δ	Violation	The interdiction is violated.
ϵ	Reconnaissance	The villain makes an attempt at reconnaissance.
ζ	Delivery	The villain receives information about his victim.
η	Trickery	The villain attempts to deceive his victim.
θ	Complicity	Victim submits to deception and thereby unwittingly helps his enemy.
A/a	Villainy/Lack	A member of a family is harmed by the villain, or lacks something.
B	Mediation	Misfortune or lack is made known.
C	Beginning Counteraction	The hero agrees to or decides upon counteraction.
$\uparrow$	Departure	The hero leaves home.
D	Donor Encounter	The hero is tested, interrogated, attacked, etc., which prepares the way for E.
E	Hero's Reaction	The hero reacts to the actions of the future Donor.
F	Receipt of Magical Agent	The hero acquires the use of a magical agent.
G	Transference	Hero is led to the whereabouts of an object of search.
H	Struggle	The hero and the villain join in direct combat.
I	Victory	The villain is defeated.
J	Branding	The hero is branded.
K	Tension Liquidated	The initial misfortune or lack is liquidated.
$\downarrow$	Return	The hero returns.
Pr	Pursuit	The hero is pursued.
Rs	Rescue	Rescue of the hero from pursuit.
o	Unrecognized Arrival	Unrecognized, he arrives home or in another country.
L	Unfounded Claims	A false hero presents unfounded claims
M	Difficult Task	A difficult task is proposed to the hero.
N	Solution	The task is resolved.
Q	Recognition	The hero is recognised.
Ex	Exposure	The false hero or villain is exposed.
T	Transfiguration	The hero is given a new appearance
U	Punishment	The villain is punished.
W	Reward	The hero is married and ascends the throne, or receives money.

Table 1: Propp's 31 functions.

4 Corpus

We demonstrate our method on the so-called *extended* ProppLearner corpus (Jahan et al., 2020), which is an expansion of the 16 tale ProppLearner corpus (Finlayson, 2017). This corpus comprises 46 Russian folktales originally collected in Russia in the late 1800s but translated into English, and then annotated using modern linguistic annotation methods for a variety of useful information. We used all 46 folktales for training the animacy and character detection stages, but excluded two of those texts (#16 and #17) from the archetype learning experiments due to errors in the alignment of archetype markings with referring expression annotations. To the best of our knowledge, this is the only corpus that provides gold-standard stereotypical character role annotations as well as plot function information. It also contains gold-standard annotations for referring expressions, coreference chains, animacy, and character (Jahan et al., 2018, 2020). We performed some manual correction on this corpus, primarily eliminating minor errors in the coreference chain and plot function annotation and merging coreference chains that were erroneously split. Table 2 shows various information about the corpus, focusing on both gold standard and automatically computed features of coreference chains, and also including counts of coreference chains that were marked with various stereotypical character roles.

Element Type	**Counts**	**Archetype**	**Gold**	**Auto.**
Texts	46	Hero	58	53
Tokens	1,09,120	Villain	97	72
Coreference Chains		Donor	28	21
Total	4,960	Helper	50	31
Gold Anim.	2,004	Princess	27	25
Auto. Anim.	2,225	Dispatcher	20	17
Gold Char.	564	False Hero	2	2
Auto. Char.	534	Others	282	313
Arch.	194			

Table 2: Counts of different archetypes of the gold-standard annotation and the automated output of the animacy-character-archetype model.

5 Approach

Our approach assumes we begin with coreference chain annotations. We first detected the animate entities using an existing state-of-the-art animacy detector (Jahan et al., 2018), then identified which of those animate entities are characters using an existing character identifier (Jahan et al., 2020). Finally, we implemented k-means clustering to learn stereotypical roles of those characters.

5.1 Animacy Detection

According to the operational definition of character found in Jahan et al. (2020), a character must be an animate object that is important to the plot. Thus the first step of role learning is to detect the animate entities. We used the animacy classifier described in Jahan et al. (2018) for animacy detection over coreference chains. We used their best-performing model (0.90 F_1), a hybrid model incorporating supervised machine learning and hand-built rules.

5.2 Character Identification

For identifying characters, we used the character identifier and the gold-standard character annotation of (Jahan et al., 2020). The character model is a supervised machine learning model that includes seven features, and it performs quite well on the extended ProppLearner corpus (0.88 F_1).

5.3 Role Clustering

To cluster identified characters into Propp's stereotypical character roles groups, we used k-means clustering[2]. Although Propp identifies seven roles we excluded the *False Hero* characters from the data because there are only two examples. We have added an extra label named *Others* which represents non-archetype characters or non-major characters. We explored different features (computed for each character) as follows.

tf-idf: We computed *tf-idf* vectors over words of the heads of the coreference chains as a feature. The vector size is 319, which means 319 unique words where each coreference chain has non-zero *tf-idf* entries for at least one place in the vector or possibly more, depending on the number of words in the head.

Bag-of-words: We computed bag-of-words vectors over coreference chain head words as a feature. The vector length is 319, one entry for each unique word across the co-reference chain heads.

Hashing: We calculated hashing vectors to convert the words of the coreference chain heads to a sparse matrix of token occurrence counts.

We explored six different vector encodings of how characters participated in plot functions (P1c, P1b, P2c, P2b, P3, and P4). Vectors P1 and P2 were computed in one of two ways: "count" where each index represents how many times a character participates in a particular function, and "binary" where each index represents whether or not a character participates a particular function.

P1c and **P1b:** These feature vectors are of length 31 (one for each of Propp's plot functions), and encodes whether there is a string match between the input character chain and the sentences containing the plot function events. We calculated this feature in both "count" (P1c) and "binary" (P1b) ways. This feature vector is intended to capture whether a character participates in a function.

P2c and **P2b:** These feature vector are of length 62 (two places for each of Propp's plot functions), and encodes whether there is a string match between the input character chain and the agent or patient arguments (computed via a semantic role labeler) for the verb associated with each plot function. We calculated this feature in both ways, "count" and "binary". These feature vectors are intended to capture whether a character participates in a function but distinguish between agent and patient participation.

P3: This feature vector is of length 62 and is a function of P2c and P2b. The first 31 places encode the difference between the P2c agent and P2c patient counts for each plot function: i.e., P3$[i]_{0-30} =$ P2c$[i]-$P2c$[i+31]$. The second 32 places encodes the P2 binary agent entry OR'd with the binary patient entry for each plot function: i.e., P3$[i]_{31-61} =$ P2b$[i] \vee$ P2b$[i{+}31]$. This feature vector is intended to capture how much more a character participates in a function as agent or patient.

P4: This feature vector is the same as P3 except the first 31 places are mapped via the $sgn()$ function to -1, 0, or 1. This feature vector captures merely whether a character on balance participates in a function more as agent or patient.

5.4 Cluster Evaluation Method

Because the output of the k-means clustering is just a set of clusters, to evaluate against the gold standard we must assign a stereotypical character role to each cluster. To do so, we followed the following procedure: (a) Order the list of seven stereotypical character role labels by their gold-standard anno-

[2] K-means (from sklearn Python library) parameters were set at $init = k-means++$, $n\_init = 10$, $max\_iter = 500$, $verbose = 0$, $random\_state = 3425$

Features	Hero	Villain	Donor	Helper	Princess	Dispatcher	Other	ARI	F_1
tf-idf	0.35	0.42	0.09	0.33	**0.70**	**0.55**	0.03	0.11	0.24
P2c	0.03	0.04	0.00	0.18	0.04	0.24	0.61	0.09	0.41
P2b	0.00	0.30	0.32	0.00	0.26	0.28	0.65	0.18	0.43
P4	0.24	0.38	0.05	0.21	0.46	0.09	0.63	0.14	0.47
P3	0.03	0.04	0.00	0.20	0.06	0.24	0.61	0.09	0.41
P1c	0.15	0.05	0.00	0.05	0.00	0.32	0.60	0.07	0.39
P1b	0.38	0.59	**0.44**	0.37	0.27	0.00	**0.69**	0.27	0.50
P1b, *tf-idf*, BOW, H	**0.60**	**0.63**	0.27	**0.37**	0.67	0.40	0.68	**0.29**	**0.58**

Table 3: Performance of the different feature sets for $k = 7$. ARI = Adjusted Rand Index, BOW = bag-of-words, H = Hashing

tation counts in descending frequency. (b) Pop the first label from the list and compute the F_1 of that specific label in each cluster based on the gold-standard annotations for characters. (c) The cluster with the maximum F_1 for that label will be assigned to that label. (d) Repeat steps $b - c$ until the label list is empty. We explored variations of this procedure using counts and percentages instead of F_1, but the final result was unchanged.

6 Results and Discussion

For each feature set explored, we swept the number of clusters (k) from 1 to 20, calculating the overall F_1 across the clustering as an objective measure. In most cases, $k = 7$ produces the highest performance, which matches the number of labels in the set. In general, the plot function P1b feature outperformed all of the other plot function features. Our model achieved the best performance (F_1 0.58) for the feature set of P1b, *tf-idf*, bag-of-words, and hashing for all clustering assignment methods.

For the case of individual cluster results, we can see that the results of *Hero*, *Villain*, *Princess*, and *Other* clusters are better than *Donor*, *Helper*, and *Dispatcher* clusters. We hypothesize that this is due to both lack of data for the latter labels, as well as lack of distinctiveness in the distributions of their plot function participation.

Donor does poorly on *tf-idf* while Princess is high, we suspect because Donor is mostly dependent on actions, not on the content of the coreference chains. On a different note, the features containing thematic information do not impact most of the classes except for Dispatcher.

7 Error Analysis

A detailed error analysis of the results revealed some minor problems for the model that depends on the external tools we have used and the quality of the data. First, the model uses the output of the animacy and character models. Therefore, our clustering model compounds the errors from those steps. Improvement in animacy and character models can improve the performance. Second, the quality of coreference chains and plot functions is critical for the model. Initially runs of our model did not achieve good performance, but the performance increased when we discovered and corrected a number of errors in the coreference chains and plot function annotations. After including missing plot function annotations and completing incomplete coreference chains, the F_1 improved from 0.45 to 0.50 for P1b. Third, some roles are not involved in very many plot functions; therefore, the model has difficulty clustering them correctly. Finally, a few characters have multiple roles simultaneously, but our model can learn only assign one role for each character. Future work might address this issue through a hierarchical clustering method that supports multiple roles simultaneously.

8 Contributions

We have made two major contributions in the area of stereotypical character role learning. First, we designed and developed a pipeline to learn stereotypical roles automatically. Second, we showed that plot functions, agent, and patient information are necessary to cluster similar roles. We provide our code and data for reproducibility of the work [3].

Acknowledgements

This work was supported in part by NSF CAREER Award IIS-1749917.

References

Meyer Howard Abrams and Geoffrey Harpham. 2014. *A glossary of literary terms*. Cengage learning.

[3] https://doi.org/10.34703/gzx1-9v95/DD6SEN

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361, Sofia, Bulgaria. Association for Computational Linguistics.

David Bamman, Ted Underwood, and Noah A Smith. 2014. A bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 370–379.

Jayadev Bhaskaran and Isha Bhallamudi. 2019. Good secretaries, bad truck drivers? occupational gender stereotypes in sentiment analysis. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 62–68, Florence, Italy. Association for Computational Linguistics.

R.H.-G. Chen, C.-C. Chen, and C.-M. Chen. 2019. Unsupervised cluster analyses of character networks in fiction: Community structure and centrality. *Knowledge-Based Systems*, 163:800–810.

Mark A. Finlayson. 2017. ProppLearner: Deeply Annotating a Corpus of Russian Folktales to Enable the Machine Learning of a Russian Formalist Theory. *Digital Scholarship in the Humanities*, 32(2):284–300.

Pablo Gervás. 2013. Propp's Morphology of the Folk Tale as a Grammar for Generation. In *2013 Workshop on Computational Models of Narrative*, volume 32 of *OpenAccess Series in Informatics (OASIcs)*, pages 106–122, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Adrian Groza and Lidia Corde. 2015. Information retrieval in folktales using natural language processing. In *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 59–66. IEEE.

Harryizman Harun and Zulikha Jamaludin. 2016. An identification of dramatis personae distribution in malaysian folktales for structural classification as a preservation means of malaysian folktales. *Revista Tecnica De La Facultad De Ingenieria Universidad Del Zulia (Technical Journal of the Faculty of Engineering, TJFE)*, 39(9):22–30.

Labiba Jahan, Geeticka Chauhan, and Mark Finlayson. 2018. A new approach to Animacy detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1–12, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Labiba Jahan, Rahul Mittal, W. Victor Yarlott, and Mark Finlayson. 2020. A straightforward approach to narratologically grounded character identification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6089–6100, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Vladimir Propp. 1968. *The Morphology of the Folktale (2nd ed.)*. University of Texas Press, Austin, TX.

Ruth Anne Robbins. 2005. Harry potter, ruby slippers and merlin: Telling the client's story using the characters and paradigm of the archetypal hero's journey. *Seattle UL Rev.*, 29:767.

Jonathan P Rowe, Eun Young Ha, and James C Lester. 2008. Archetype-driven character dialogue generation for interactive narrative. In *International Workshop on Intelligent Virtual Agents*, pages 45–58. Springer.

Marcin Skowron, Martin Trapp, Sabine Payr, and Robert Trappl. 2016. Automatic identification of character types from film dialogs. *Applied Artificial Intelligence*, 30(10):942–973. PMID: 29118463.

Josep Valls-Vargas, Santiago Ontanón, and Jichen Zhu. 2014a. Toward automatic character identification in unannotated narrative text. In *the 7th Intelligent Narrative Technologies Workshop (INT7)*, pages 38–44, Milwaukee, WI.

Josep Valls-Vargas, Jichen Zhu, and Santiago Ontanón. 2014b. Toward automatic role identification in unannotated folk tales. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 10.

Josep Valls-Vargas, Jichen Zhu, and Santiago Ontañón. 2016. Predicting proppian narrative functions from stories in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 12.