6-15-2021

Inducing Stereotypical Character Roles from Plot Structure

Labiba Jahan
Florida International University, ljaha002@fiu.edu

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

INDUCING STEREOTYPICAL CHARACTER ROLES FROM PLOT STRUCTURE

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Labiba Jahan

2021

To: John L. Volakis
College of Engineering and Computing

This dissertation, written by Labiba Jahan, and entitled Inducing Stereotypical Character Roles from Plot Structure, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Christine Lisetti

Anthony Dick

Santiago Ontañón

Monique Ross

Mark A. Finlayson, Major Professor

Date of Defense: June 15, 2021

The dissertation of Labiba Jahan is approved.

John L. Volakis
College of Engineering and Computing

Andrés G. Gil
Vice President for Research and Economic Development
Dean of the University Graduate School

Florida International University, 2021

DEDICATION

To my mother Fahmida Banu Quraishi and my three sisters Rowshon Jahan, Nilufa Yeasmin, and Farhana Jahan. I wouldn’t be the person I am today without your love and support, my four precious women.

ACKNOWLEDGMENTS

I want to express my gratitude to my advisor, Dr. Mark A. Finlayson; without his encouragement and guidance, I would not have been able to complete my Ph.D. journey. He taught me a lot of skills and techniques for succeeding academically and in real life. I appreciate his enormous efforts in preparing me for my future endeavor. I would also like to thank my committee members, Dr. Christine Lisetti, Dr. Anthony Dick, Dr. Santiago Ontañón, and Dr. Monique Ross. I appreciate the time and valuable advice I received from them throughout my Ph.D. I am grateful to Geeticka Chauhan and Rahul Mittal, two undergraduate students I mentored; they assisted me in different parts of my thesis work. I am especially thankful to my colleague and friend, W. Victor H. Yarlott, for his help and support in my work. I am also thankful to my colleagues and friends from Cognac Lab including Josh, Deya, Muhammed, Anurag, Mustafa, Mireya, Samira, Andres, and Armando. I am grateful to my partner, S. M. Asifur Rahman, for being by my side and supporting me in chasing my dream. I also like to thank my brother-in-laws Md. Mohidur Rahman, Masud Rana, and Md. Kamruzzaman, sister-in-law Adbana Anis, father-in-law S. M. Anisur Rahman, mother-in-law Nasima Islam, for their love and support. A special thanks to my nephews and nieces, Rajit, Megh, Zain, Joyee, Manha, and Audri. I want to thank my friends Misty, Banya, Drishty, Rimon, Biswa, Omit, Mimi, Orny, Tania, Tumpa, Orika, Riva, Farzana, Masud, Naznin, Shagufta, Tani, Akash, Proyash, Prianka, Nusrat, Saki, Maliha, Tamanna, Ishra, and Joyee for everything they have done for me. I am also thankful to my cousin Masudul Hassan Quraishi for his help and support. Last but not least, I am grateful to my father, A. H. M. Easin, for dedicating his entire life to his family.

ABSTRACT OF THE DISSERTATION

INDUCING STEREOTYPICAL CHARACTER ROLES FROM PLOT STRUCTURE

by

Labiba Jahan

Florida International University, 2021

Miami, Florida

Professor Mark A. Finlayson, Major Professor

If we are to understand stories, we must understand characters: characters are central to every narrative and drive the action forward. Critically, many stories (especially cultural ones) employ stereotypical character roles in their stories for different purposes, including efficient communication among bundles of default characteristics and associations, ease understanding of those characters' role in the overall narrative, and many more. These roles include ideas such as hero, villain, or victim, as well as culturally-specific roles such as, for example, the donor (in Russian tales) or the trickster (in Native American tales). My thesis aims to learn these roles automatically, inducing them from data using a clustering technique.

The first step of learning character roles, however, is to identify which coreference chains correspond to characters, which are defined by narratologists as animate entities that drive the plot forward. The first part of my work has focused on this character identification problem, specifically focusing on the problem of animacy detection. Prior work treated animacy as a word-level property, and researchers developed statistical models to classify words as either animate or inanimate. I claimed this approach to the problem is ill-posed and presented a new hybrid approach for classifying the animacy of coreference chains that achieved state-of-the-art performance.

The next step of my work is to develop approaches first to identify the characters and then a new unsupervised clustering approach to learn stereotypical roles. My character

identification system consists of two stages: first, I detect animate chains from the coreference chains using my existing animacy detector; second, I apply a supervised machine learning model that identifies which of those chains qualify as characters. I proposed a narratologically grounded definition of character and built a supervised machine learning model with a small set of features that achieved state-of-the-art performance.

In the last step, I successfully implemented a clustering approach with plot and thematic information to cluster the archetypes. This work resulted in a completely new approach to understanding the structure of stories, greatly advancing the state-of-the-art of story understanding.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1

INTRODUCTION

1.1 Motivation

Characters are an indispensable element of the narrative. Most definitions of narrative acknowledge the central role of character: Monika Fludernik, a famous narratologist, defines a narrative as "a representation of a possible world . . . at whose centre there are *one or several protagonists* of an anthropomorphic nature . . . who (mostly) perform goal-directed actions . . . " [Fludernik, 2009, p. 6]. Thus, if we are to achieve the long-term goal of automatic story understanding, it is critical that we be able to automatically identify a story's characters, distinguishing them from non-character entities such as props, locations, or other referents, and further understand their function in the story, which often falls into one or more stereotypical roles. The goal of my thesis is to learn these roles automatically, inducing them from data using a novel co-clustering technique.

My thesis will have an impact on Natural Language Processing, Story Understanding, and Cognitive Science because stories are filled with stock or stereotypical characters (such as heroes, villains, victims, tricksters, and so forth) that allow efficient communication of cultural and situational knowledge. If we wish to enable culturally sensitive story understanding, our systems will naturally have to be aware of cultural roles that are played, both in stories or in society or culture at large. How does one learn these categories? One can be explicitly taught them, of course, but there is evidence from anthropology and psychology that people actually learn these categories in an unsupervised fashion, inferring them from their repeated occurrence across the many stories that they hear day-in and day-out. My work seeks to model this process computationally, producing a system that will learn these character roles automatically in a context-sensitive way. It will directly advance our understanding of narratives, as well as the human language.

1.2 Problem Statement and Research Components

The research problem that I solved is learning stereotypical roles from narratives in an unsupervised fashion. It consists of three major research components: animacy detection, character identification, and stereotypical role learning.

1.2.1 Component 1: Animacy Detection

The first step toward character detection is animacy detection, where animacy is the characteristic of being able to independently carry out actions in a story world (e.g., movement or communication). All characters are necessarily animate—although not all animate things are necessarily characters—and so detecting animacy will immediately narrow the set of possibilities for character detection. Prior work has conceived of animacy as a word-level phenomenon, marking animacy as an independent feature on each individual word (e.g., [Orăsan and Evans, 2007], [Bowman and Chopra, 2012], [Karsdorp et al., 2015]). However, characters and other entities are expressed in texts as coreference chains made up of referring expressions [Jurafsky and Martin, 2007], where referring expressions are natural language expressions used to perform reference, and coreference chain is a set of co-referring expressions. So, we need some way of computing animacy on the chains directly. We can attempt to compute animacy directly on the referring expressions and coreference chains, which is the approach I have pursued in my work. I developed a hybrid system merging supervised machine learning (ML) and a small number of hand-built rules to compute the animacy of referring expressions and coreference chains. This method achieved state-of-the-art performance [Jahan et al., 2018].

1.2.2 Component 2: Character Identification

The second step is to identify which animate entities correspond to characters (those important entities that help drive the plot forward). Numerous prior approaches have incorporated character identification in one way or another. Some approaches, e.g., examining charaters' social networks [Sack, 2013], take character identification for granted, implementing heuristic-driven identification approaches over named entities or coreference chains that are not examined for their efficacy. Other approaches have sought to solve the character identification task specifically, but have relied on domain-specific ontologies [Declerck et al., 2012] or complicated case bases [Valls-Vargas et al., 2014a]. Others have taken supervised machine learning approaches [Calix et al., 2013]. Regardless, all of the prior work has, unfortunately, had a relatively impoverished view of what a character is, from a narratological point of view. The authors had either previous knowledge of the character in their models or considered animate entities and name entities as characters in those works. In particular, a key aspect of any character is that it *contributes to the plot*; characters are not just any animate entity in the narrative. I proposed a narratologically grounded definition of character based on its participation in the plot events. Also, I implemented a preliminary supervised machine learning model with a small set of features that achieved state-of-the-art performance [Jahan et al., 2020a].

1.2.3 Component 3: Stereotypical Role Learning

The third step, which comprised the remainder of my dissertation, is to learn the stereotypical roles the characters play because, without the ability to understand that a character is a hero (rather than a villain, or some other role), no computer will ever have an understanding of a story in the same way that a person does. Few models are built to solve this task, but most of them incorporated some previous knowledge of stereotypical roles in the models; as an example, some of them used computer-aided qualitative data analysis software [Harun and Jamaludin, 2016], some of them included archetype information in an ontology [Groza and Corde, 2015], others used feature vectors of archetype information [Valls-Vargas et al., 2016]. I proposed a new approach to learn stereotypical roles in an unsupervised way and implemented a clustering approach to learn them using a character's plot and thematic information that performed well on Russian folktales.

1.3 Dissertation Contributions

My dissertation made several contributions in the cases of the three research components I described above.

Animacy Detection

I made five major contributions in the area of animacy detection. First, I have redefined the problem of animacy classification as one of marking animacy on coreference chains, in contrast to all prior work that seeks to mark the animacy at the word level. Second, I have presented a hybrid system merging an SVM classifier and hand-built rules to predict the animacy of referring expressions directly, achieving performance of 0.90 F_1, which is comparable to the state of the art for word-level animacy detection [Jahan et al., 2018]. Third, I used a majority voting approach to obtain the animacy of coreference chains. The overall performance of this approach is substantially improved in comparison with my prior work [Jahan et al., 2017]. Fourth, I provided 15 texts annotated for word-level animacy and 142 texts annotated for coreference chain animacy, as well as the code reproducing the results. Finally, I tested and confirmed the generilizability of my proposed animacy models. Additionally, I published code and data for the community (link: `https://dspace.mit.edu/handle/1721.1/116172`).

Character Identification

I made four major contributions in the area of character identification. First, I proposed a more appropriate definition of *character*, contrasting with prior computational works which did not provide a theoretically grounded definition of character based on its participation in the plot. Additionally, I reported the findings of a review of the literature that is helpful to delineate and define the concept of character (§3.3). Second, I annotated 170 texts for character, generating data that will be useful for the community. Third, I have demonstrated a supervised machine learning classifier for identifying characters, achieving weighted average of 0.90 F_1, establishing a new standard for this task [Jahan et al., 2020a]. Finally, I tested and confirmed the generilizability of my proposed character model and I published my code and data for the community (link: `https://doi.org/10.34703/gzx1-9v95/RB6ZH0`).

Stereotypical Role Learning

I made two major contributions in the area of stereotypical role learning. First, I designed and developed a pipeline to automatically learn stereotypical roles. Second, I proved that plot functions and thematic role information are important to cluster similar archetypes. Moreover, I will publish code and data for the community.

During my work I mentored two undergraduate students, Geeticka Chauhan and Rahul Mittal who helped me in some parts of the data annotation, features extraction and running experiments.

1.4 Outline

The dissertation proceeded as follows. First, I discussed the details of my animacy model; motivation, data and annotation, methodology, results, etc. (§2). I next described the details of my character model just like I did for the animacy work (§3), following which I discussed the details of my stereotypical role learning model (§4). Although I have discussed prior work briefly in the motivation section of each chapter, I summarized work related to this study (§5) before I concluded (§6).

CHAPTER 2

ANIMACY DETECTION

2.1 Motivation

Animacy is the characteristic of being able to independently carry out actions (e.g., movement, communication, etc.). For example, a person or a bird is animate because they move or communicate under their own power. On the other hand, a chair or a book is inanimate because they do not perform any kind of independent action. Animacy is a useful semantic property for different NLP systems, including word sense disambiguation (WSD), semantic role labeling (SRL), coreference resolution, among many others. Animacy can be used to distinguish different senses and thus help a WSD system assign senses to different words. As an example, animacy has been applied in grouping senses from WordNet [Palmer et al., 2004, 2007]. Animacy can also be used directly in a WSD system to decide thematic assignment, which is useful for assigning senses: for example, Carlson and Tanenhaus [1988] used the presence of an animate subject in a sentence to determine if a the verb is transitive, which is useful for thematic role assignment. Another task where animacy can play an important role is semantic role labeling (SRL). Agentive or semantic subject roles must often be filled by animate entities, whereas goal, theme, patient, instrument and location roles are often filled by inanimate entities [Kittilä et al., 2011]. In some works [Connor et al., 2013, Kittilä, 2006, for example], animacy is used as a feature that helps to identify agents, and Ferreira [1994] showed how knowing the animacy of roles allows one to better identify the passive voice. In many coreference resolution systems [Raghunathan et al., 2010, Iida et al., 2003, Cardie and Wagstaf, 1999, for example] animacy is used as a semantic feature to determine co-referents of an expression.

In addition to these broad uses of animacy, I am particularly interested in detecting animacy with a view toward identifying characters in stories. Most definitions of narrative acknowledge the central role of character, for example: “a representation of a possible world ... at whose centre there are one or several protagonists of an *anthropomorphic* nature ... who (mostly) perform goal-directed actions ...” *(emphasis ours)* [Fludernik, 2009, p. 6]. If we are to achieve the long-term goal of automatic story understanding, it is critical that we be able to automatically identify a story’s characters, distinguishing them from non-character entities. All characters are necessarily animate—although not all animate things are necessarily characters—and so detecting animacy will immediately narrow the set of possibilities for character detection.

2.2 Approach

Prior work treated animacy as a word-level phenomenon, marking animacy as an independent feature on individual words [Orăsan and Evans, 2007, Bowman and Chopra, 2012, Karsdorp et al., 2015]. But word-level animacy is not always sufficient to identify an animate or an inanimate object. For example, *horse* is normally animate, but a *dead horse* is obviously inanimate. On the other hand, *tree* is an inanimate word but a *talking tree* is definitely an animate thing. So, assigning animacy at the word level confuses the issue and makes it more difficult to classify these type of complex cases.

Furthermore, referents are expressed in texts as coreference chains comprised of referring expressions, and so conceiving of animacy as a word-level phenomenon requires an additional method for computing chain animacy from word animacy. One way to do this is to combine word-level animacy markings—say, using majority vote—into referring expressions animacy and then coreference chains. As it turns out, this does not work all that well and I used this method as my baseline. Alternatively, we can attempt to compute animacy directly on the referring expressions and then use majority vote of referring-expression-level animacy to compute animacy of coreference chains, the approach I pursued here.

Although detecting animacy might seem to be straightforward, it presents a number of subtleties. For example, some theorists have proposed closed lists of linguistic expressions that should be automatically considered animate entities, such as titles, animals, or personal pronouns [Quirk et al., 1985, Yamamoto, 1999]. However, texts, especially stories about unreal worlds, can arbitrarily introduce characters that would not be animate in real life, for example, walking stoves or talking trees. Figure 2.1 shows an example sentence from a Russian fairytale which contains three animate chains, one of which is a tree that talks: trees would not be normally be considered animate according to canonical lists of animate entities. Therefore some context sensitivity in detection is needed.

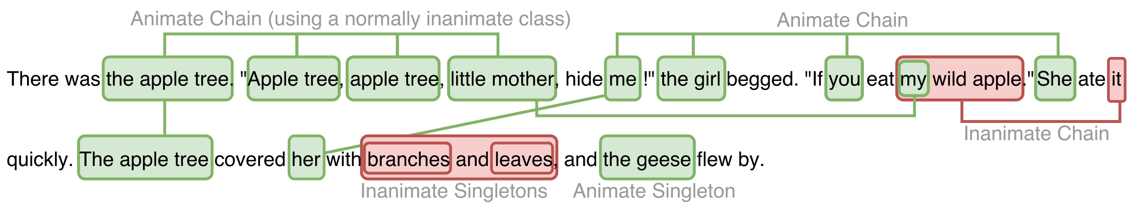

There was the apple tree. "Apple tree, apple tree, little mother, hide me !" the girl begged. "If you eat my wild apple." She ate it
quickly. The apple tree covered her with branches and leaves, and the geese flew by.

Figure 2.1: Example text containing animate and inanimate coreference chains. Colored boxes represent referring expressions, while links between them signify coreference. Animate chains are green, while inanimate chains are red. The text is drawn from Story #113 *The Magic Swan Geese* [Guterman, 1975, p. 350] and has been slightly modified for clarity.

In my work, I computed animacy directly on referring expressions, and transferred those markings up to the coreference chain level, to get a direct classification of the animacy of the whole chain. I presented a hybrid system combining statistical machine learning (ML) and hand-built rules for classifying the animacy of referring expression, and also presented a voting model to identify the animacy of coreference chains based on the animacy of the chain's constituent referring expressions.

2.3 Data and Annotation

I started this project seeking to use existing data annotated for animacy, as there have been a number of studies on animacy detection already. However, no prior data in English was readily available to use; the best performing prior work on word-level animacy was done on a corpus of 74 stories comprising 74,504 words in Dutch [Karsdorp et al., 2015]. Orăsan and Evans [2007] did their work in English but their data was not available. Therefore I sought other data (specifically stories, because of my interest in story understanding), and my annotated data was a corpus comprising a variety of Russian folktales, Islamist Extremist stories, and Islamic Hadiths that are freely available and assembled for other work, and had been annotated for referring expressions and coreference chains [Finlayson, 2017, Finlayson et al., 2014]. The composition of the corpus is shown in Table 2.1.

Text Types	# Texts	# Tokens	# Ref. Exp.	# Coref. Chains
The extended ProppLearner	46	109,120	20,391	4,950
Islamist Extremist Texts	32	26,557	8,041	3,684
Islamic Hadiths	64	20,477	6,266	2,307
Total	142	156,154	34,698	10,941

Table 2.1: Counts of various text types in the three corpora. Ref. Exp. stands for referring expression and Coref. stands for coreference.

The corpus contains 46 Russian folktales, originally collected in Russian in the late 1800's but translated into English in the mid-twentieth century [Finlayson, 2017]. The other portion (the N2 corpus) contains 96 stories of relevance to Islamist Extremists [Finlayson et al., 2014]. All but 31 of the texts in the corpus already contained gold-standard annotations for token and sentence boundaries, parts of speech, referring expressions, and coreference chains (as well as other layers of annotation. I processed these 31 un-annotated texts using the Stanford CoreNLP suite [Manning et al., 2014], automatically generating tokens, sentences, parts of speech, referring expressions, and coreference chains.

	Total Entity	**Ani. Entity**	**Inani. Entity**	**Unique Ani.**	**Unique Inani.**
Token (15 stories)	23,291	3,896	19,395	291	2,221
Referring Expression (142 stories)	34,698	22,052	12,646	1,104	2,249
Coreference-chain (142 stories)	10,941	3,832	7,109	-	-

Table 2.2: The total number of animate and inanimate tokens, referring expressions, and coreference chains, with breakdowns of each class's number of unique entities. Ani. Entity and Inani. Entity stand for the total number of animate and inanimate entities; Unique Ani. and Unique Inani. stand for the total number of unique animate and unique inanimate entities.

I along with my mentee, Geeticka Chauhan, singly and doubly annotated the corpus for animacy of coreference chains, and the first fifteen stories for animacy at the word level. We propagated the animacy annotations from the chains to their constituent referring expressions to generate animacy annotations at that level. Because I had to automatically compute referring expression and coreference chains on 31 of the texts, and the CoreNLP coreference resolution is somewhat noisy, I hand-corrected the chains. I did this hand-correction using the Story Workbench annotation tool [Finlayson, 2008, 2011] that allows for the manipulation and correction of referring expression and coreference chains.

Gold Standard Corpora is the standard collections of corpora that are verified by a meaningful annotation evaluation method [Wissler et al., 2014]. The annotation of the animacy of coreference chains and referring expressions for the first fifteen stories was performed by me and Geeticka Chauhan. Disagreements were discussed and corrected to generate a gold-standard annotation. Agreement for the coreference-level was 0.99 F_1 and 0.99 Cohen's kappa coefficient (κ), which represents near-perfect overall agreement

[Landis and Koch, 1977]. The annotation of the rest of the stories was performed by only me.

Referring Expression	**Class**	**Explanation**
the dragon, Abu Bakr	Ani.	Normally ani. entities
walking stove, talking tree	Ani.	Normally inani. but are ani. in context
"what it is"	Inani.	Discourse acts, when marked as referents
the mosque, this world	Inani.	Normally inani. objects
dead horse	Inani.	Normally ani. but are inani. in context
her eyes, his hands	Inani.	Inani. parts of ani. entities
Word		
princess, dragon	Ani.	Nouns denoting ani. entities
he, she, her	Ani.	Personal pronouns referring to ani. objects
stronger [dragon]	Ani.	Adjectives that suggest animacy
Morning, [talking] stove	Ani.	Usually inani. but are ani. in context
Kiev, world	Inani.	Nouns denoting inani. entities
it, that	Inani.	Personal pronouns referring to inani. objects

Table 2.3: Examples of annotation of coreference- and word-level animacy. At the word level, only an adjectives suggesting animacy or nouns referring to an animate object are marked animate. Everything else (including verbs, adverbs, determiners, and so forth) are marked inanimate. Ani. stands for Animate and Inani. stands for inanimate.

We also annotated the first fifteen Russian tales for word-level animacy so that I could test via re-implementation the existing best performing word animacy model [Karsdorp et al., 2015]. This annotation was done under the following guidelines. First, all nouns that would refer to animate entities in real life, such as humans or animals, as discussed in [Quirk et al., 1985, pp. 314 & 345] were marked animate. We marked gendered pronouns as animate, e.g., *he*, *she*, *his*, *hers*, etc. We also marked adjectives suggesting animacy as animate, e.g., *alive*, *vital*, *kindlier*, etc., whereas adjectives implying inanimacy, such as *dead* in the noun phrase *dead horse*, were marked inanimate. Second, we marked as animate any words directly referring to entities that acted animately in a story, regardless of the default inanimacy of the words. For example, we marked *stove* animate in the case of a walking stove, or *tree* animate in the case of a talking tree. This also covered proper names that might normally be marked as inanimate because of their ostensible class, such as those underlined in the next example:

> *All of them were born in one night—the eldest in the evening, the second at midnight, and the youngest in the early dawn, and therefore they were called Evening, Midnight, and Dawn. [Guterman, 1975, Tale #140, p. 458]*

The word-level annotation was done by me and Geeticka Chauhan. Disagreements were discussed and corrected to generate a gold-standard annotation. We annotated every word in the corpus for animacy directly (marking each word as either animate or not). Agreement was 0.97 F_1 and 0.97 Cohen's kappa coefficient (κ), which represents near-perfect overall agreement [Landis and Koch, 1977].

A summary of the counts of animate and inanimate words, referring expressions, and coreference chains is given in Table 2.2. Examples of animate and inanimate words are given in Table 2.3.

2.4 Methodology

My hybrid system is comprised of two parts: a rule-based classifier that can mark the animacy of roughly 50% of the referring expressions, followed by a statistical classifier trained on the annotated data that can be applied to the remaining referring expressions. Once all referring expressions are marked for animacy, the animacy of a coreference chain is inferred from the animacy of its constituent referring expression.

2.4.1 Rules

I implemented five rules that considered semantic subjects parsed from the semantic role labeler associated with the Story Workbench annotation tool [Finlayson, 2008, 2011], the named entities computed using the classic API of Stanford dependency parse [Manning et al., 2014, v3.7.0], and knowledge from WordNet [Fellbaum, 1998]. These rules were inspired by existing rule-based animacy systems. I also considered the last word of a referring expression in most of the rules because it helps to mark quotes as inanimate, as well as to detect the regular animate and inanimate referring expression.

1. If the last word of a referring expression is a gendered personal, reflexive, or possessive pronoun (i.e., excluding *it*, *its*, *itself*, etc.), the model marked it animate.
2. If the last word of a referring expression is the semantic subject to a verb, the model marked it animate.
3. If a referring expression contains a proper noun the model marked it animate. I excluded anything tagged as *location*, *organization*, or *money*, as determined by the Stanford CoreNLP NER system.
4. If the last word of a referring expression is a descendant of *living_being* in WordNet, the model marked it animate.
5. If the last word of a referring expression is a descendant of *entity* WordNet, the model marked it inanimate.

2.4.2 Features

I explored seven different binary and vector features to train the statistical classification model, some of which are drawn from prior work.

Word Embeddings (WE)

I computed pre-trained word embeddings in 300 dimensions for all the words in the stories using the skip-gram architecture algorithm [Mikolov et al., 2013]. I used the DeepLearning4J library [Deeplearning4j Development Team, 2017], and configured the built-in skip-gram model with a minimum word frequency of 3, layer width (dimensions) of 300, a window size of 5, and trained for 10 iterations. I explored a few different combinations of these parameters, but found that these settings produced the best results. This is a vector feature drawn from [Karsdorp et al., 2015], and is primarily relevant to classifying word-level animacy. I ran this model on each word of our data and used the output vector as a feature.

Word Embeddings on Referring Expressions (WER)

I calculated pre-trained word embeddings in 450 dimensions for just the words within the referring expressions, again using the skip-gram approach as above, except with a minimum word frequency of 1 (this is a vector feature). This approach worked better for 450 dimensions (rather than 300), which I discovered after exploring the parameter value from 50-600. I ran this model on each referring expression of our data and used the output vector as a feature.

Composite Word Embedding (CWE)

I computed a composite pre-trained word embedding for the neighborhood (three words before and three words after) of each word, adding the word embedding vectors for three words before and three words after the target word (excluding the target). This is also a vector feature and is again partially drawn from [Karsdorp et al., 2015]. The idea of this feature is that it estimates the similarities of the context among all animate words (or all inanimate words) as well as the dissimilarities of animate from inanimate, and vice versa.

Parts of Speech (POS)

By analogy with the other embeddings, I computed an embedding over part of speech tags in 300 dimensions, with the same settings as in feature #1 (WE). This feature models the tendency of nouns, pronouns, and adjectives to refer to animate entities.

Noun (N)

I checked whether a given referring expression contained a noun and encoded this as a boolean feature because I observed that in the first 15 stories 43% of nouns are animate. Thus this feature explicitly captures the tendency of nouns to refer to animate entities. I used dependency parses generated by the classic API of Stanford dependency parser [Manning et al., 2014, v3.7.0].

Grammatical Subject (GS)

Animate references tend to appear as the grammatical subjects of verbs [Ovrelid, 2005]. I used dependency parses generated by the classic API of Stanford dependency parser [Manning et al., 2014, v3.7.0] to check if the last word of a given referring expression was used as a grammatical subject relative to any verb in the sentence, and encoded this as a boolean feature.

Semantic Subject (SS)

I also computed whether or not a referring expression appeared as a semantic subject to a verb. I used the semantic role labeler associated with the Story Workbench annotation tool [Finlayson, 2008, 2011] to compute semantic roles for all the verbs in the stories. I then checked whether the last word of a given referring expression contained an ARG0 for a verb (an exact match was not required), and encoded this as a boolean feature.

2.4.3 Classification Models

I implemented the classification models using SVM [Chang and Lin, 2011], with a Radial Basis Function Kernel. The features used to train the different models are shown in Table 2.6. I trained each model using cross validation, and report macroaverages across the performance on test folds. I have three models for animacy: referring expressions, coreference chains, and words. For the referring expression animacy model, I implemented three approaches. The first is a ML-only approach, in which I explored different combinations of features: word embedding over referring expressions (WER), noun (N), grammatical subject (GS), and semantic subject (SS). I configured the SVM with $\gamma = 1$, $C = 0.5$ and $p = 1$. I measured the performance of the classifier using 10-fold cross validation. The second approach is a rule based system and the third approach is a hybrid system where I first applied the rules, then applied the ML classifier for referring expressions not covered by the rules.

For the coreference chain animacy model, I implemented a majority voting approach for combining the results of the referring expression animacy model to obtain a coreference animacy prediction. In the case of ties, the chain was marked inanimate.

2.5 Preliminary Analysis

This is a preliminary study, and I only use a small corpus of 15 folktales to demonstrate the feasibility of the approach. I first annotated animacy on coreference chains directly, and then propagated these markings to the referring expressions. Using these annotations I then trained a support vector machine (SVM) classifier for the animacy of referring expressions themselves, and compared two methods for computing the animacy of a coreference chain using those values. Majority voting performed best in this context, and it outperforms a baseline that computes referring expression animacy by majority vote over the word-level animacy markings. Overall I built three different models for animacy detection. The first is the referring expression model, on which the second model for coreference chains builds. I also built a third model for word-level animacy, which is used for the baseline comparison.

2.5.1 Data

I used a small set of the ProppLearner corpus for my preliminary study. The corpus contains 15 tales, originally collected in Russian in the late 1800's but translated into English in the mid-twentieth century. Table 2.4 summarizes counts of various aspects of the annotated data. The corpus contains gold-standard annotations for token and sentence boundaries, parts of speech (Penn Treebank II Tagset; [Marcus et al., 1993]), referring expressions, and coreference chains (as well as other layers of annotation). The annotation procedure is described in the (§2.3) section.

	Token	**Referring Expressions**	**Coreference Chains**
Total	23,291	6,631	1,633
Animate	3,896	4,288	344
Inanimate	19,395	2,343	1,289
Unique Items			
Animate	291	798	-
Inanimate	2221	1459	-
Total	2,199	2,231	-
Tokens	**Noun**	**Pronoun**	**Adjective**
Animate	1,658 (43%)	2,252 (58%)	38 (1%)
Inanimate	2,220 (11%)	401 (2%)	862 (4%)

Table 2.4: Counts of various aspects of annotated data, including total number of animate and inanimate tokens, referring expressions, and coreference chains, with breakdowns of number of unique items and part of speech in each class.

2.5.2 Models

I implemented the classification models using SVM [Chang and Lin, 2011], with a Radial Basis Function Kernel. I varied the features used to train the different models as shown in Table 2.5. I trained each model using cross validation, and report macroaverages across the performance on test folds.

I have three models for animacy: referring expressions, coreference chains, and words. For our referring expression animacy model, I explored different combinations of the features: word embedding over referring expressions (WER), noun (N), grammatical subject (GS), and semantic subject (SS). I configured the SVM with $\gamma = 1$, $C = 0.5$ and $p = 1$, which were chosen after a small amount of parameter space exploration. The first two values are relatively low in the range for these parameters, which is appropriate for a balanced class situation. I measured the performance of the classifier using 10-fold cross validation.

I calculated two baselines for referring expression animacy. The first is the majority class baseline (inanimate is the majority class). The second combines word-level animacy predictions generated by my word animacy model (discussed below) via a majority vote.

For the coreference chain animacy model, I implemented two majority vote approaches for combining the results of the referring expression animacy model to obtain a coreference animacy prediction. First, I computed the majority vote considering all referring expressions in a coreference chain. In the case of ties, the chain was marked inanimate. Because short coreference chains were responsible for much of the poor performance, I also calculated the performance of majority voting excluding chains of length four and below.

To compare with prior work, I also implemented a word animacy model, adapting an existing system with the best performance [Karsdorp et al., 2015]. That model used features based on word N-grams, parts of speech, and word embeddings. Similarly, we implemented our classifier using word embeddings over words (WE), combined word embeddings (CWE), and parts of speech (POS). The SVM was configured with $\gamma = 5$, $C = 5000$ and $p = 1$, which were chosen after a small amount of parameter space exploration. The first two values are relatively high in the range for these parameters, which is appropriate for a unbalanced class situation. I measured the performance with 10-20 fold cross-validation and mentioned the best performance with 20-fold cross-validation in Table 2.5.

2.5.3 Findings

I evaluated the models by measuring accuracy, precision, recall, F_1, and Cohen's kappa (κ) relevant to the gold-standard annotations. Table 2.5 summarizes the results for both the animate and inanimate classes.

In the case of referring expression animacy I omit some combinations of features (e.g., WER & N) that produced especially poor results. I obtained the best result using three features: word embeddings over referring expressions (WER), noun (N) and semantic subject (SS).

For the coreference animacy model, majority vote does not work as well as expected, with an overall F_1 of 0.61 when calculated over all chains. This poor performance relative to the word and referring expression animacy models is due largely to under-performance on short coreference chains (those with four referring expressions or fewer). This suggests that in future work we need to concentrate our effort on solving the short chain issue.

The word model performed very close to the prior state of the art with the small data set. My model achieved F_1 of 0.98 for the inanimate class, where the state of the art achieved 0.99. On the other hand, my model achieved an F_1 of 0.90 for the animate class, where the state of the art achieved 0.93.

				Inanimate			**Animate**		
Model	**Feature Set**	**Acc.**	κ	**Prec.**	**Rec.**	**F_1**	**Prec.**	**Rec.**	**F_1**
Word	Karsdorp et al.	-	-	0.98	0.99	0.99	0.94	0.91	0.93
	WE, CWE, POS	96%	0.87	0.98	0.98	0.98	**0.91**	**0.88**	**0.90**
Ref. Exp.	Baseline MFC	37%	0	0.38	1.0	0.55	0	0	0
	Baseline MV	75%	0.53	0.59	0.99	0.74	0.99	0.62	0.76
	WER	72%	0.49	0.58	0.99	0.73	0.98	0.57	0.72
	N	80%	0.56	0.85	0.60	0.70	0.80	0.93	0.86
	GS	80%	0.56	0.85	0.60	0.70	0.79	0.93	0.86
	SS	76%	0.51	0.67	0.74	0.70	0.83	0.78	0.80
	WER, GS	84%	0.64	0.89	0.66	0.76	0.82	0.95	0.88
	WER, SS	87%	0.72	0.87	0.79	0.82	0.87	0.91	0.89
	N, GS, SS	80%	0.56	0.84	0.60	0.70	0.79	0.93	0.86
	WER, N, GS	84%	0.64	0.88	0.67	0.76	0.82	0.95	0.88
	WER, N, GS, S	87%	0.73	0.85	0.80	0.83	0.88	0.90	0.89
	WER, N, SS	86%	0.70	0.83	0.77	0.80	**0.87**	**0.91**	**0.90**
Coref.	Maj. vote (all)	79%	0.48	0.93	0.80	0.86	0.50	0.76	0.61
	Maj. vote (long)	84%	0.68	0.86	0.78	0.82	**0.82**	**0.89**	**0.86**

Table 2.5: Preliminary Results of different Animacy Models (Bolded according to when our F_1 measure is higher). MFC stands for "Most Frequent Class", MV stands for "Majority Vote" and the other abbreviations stand for features as indicated in the "Features" section.

Nevertheless, there is no prior work that reports animacy classification results directly for referring expressions and coreference chains, and so these results set the initial foundation for animacy classification of these objects.

2.5.4 Future Direction

A detailed error analysis of the results revealed at least four major problems for the classifier which are useful for the future direction: short chains, quotations, agency selection restrictions, and proper names.

Short coreference chains

Determining the animacy of short coreference chains is apparently a challenging task for the system. As the length of a chain tends toward a single referring expression, the coreference classifier performance should converge to the referring expression classifier performance. However, for chains between two and four referring expressions long, the majority voting approach seems to fall short. I suspect this is because many referring expressions are themselves quite short, and can contain false alarms: e.g., our system classifies “his hands” as animate because of the animate word “his” in the expression. I believe one approach to solving this problem is more data, and explicitly incorporating the animacy of heads of noun phrases as features.

Quotes

The second problem is that many quotes are full of animate words, e.g., ”the fate of the tsar ’s daughter to go to the dragon” is a phrase that is itself a referring expression in one story, and should be inanimate according to my animacy annotation rule but the classifier detects it as animate because it finds three animate words ”tsar”, ”daughter” and ”dragon” in that quote. This will require some rule-based processing to address.

Selectional restrictions

A third problem is that although animacy correlates with semantic subject position, it is not strictly implied by it. Consider the difference between “The bird flew across the field” (implies that *the bird* is animate) and “The ball flew across the field” (the ball is inanimate). To address this problem, I plan to incorporate animacy selectional restrictions as training features, where the selectional restrictions are drawn from existing lexical resources (e.g., VerbNet; [Schuler, 2005]). This will allow me to distinguish between semantic roles which imply animacy and those which do not.

Names identical to inanimate entities

Finally, in the folktales we see names whose surface form are identical to inanimate entities, e.g., *Evening*, *Midnight*, or *Dawn*, as mentioned previously. Addressing this will requiring integrating named entity recognition into the system.

2.6 Results and Discussion

I calculated two baselines for referring expression animacy. The first baseline is to choose the majority class (animate). The second baseline combines word-level animacy predictions generated by my word animacy model via a majority vote; I measured the upper bound for this over the 15 texts for which I have gold-standard word animacy annotations.

				Inanimate			**Animate**		
Model	**Feature Set**	**Acc.**	κ	**Prec.**	**Rec.**	$\mathbf{F_1}$	**Prec.**	**Rec.**	$\mathbf{F_1}$
Ref. Expr.	Baseline MFC	61%	0.0	0.0	0.0	0.0	0.61	1.0	0.76
	Baseline MV	75%	0.53	0.59	0.99	0.74	0.99	0.62	0.76
	WER, N, GS, SS	76%	0.47	0.80	0.51	0.62	0.76	0.92	0.83
	N, GS	78%	0.51	0.83	0.54	0.65	0.77	0.93	0.84
	N, SS	79%	0.53	0.80	0.60	0.68	0.78	0.91	0.84
	N, GS, SS	79%	0.53	0.81	0.59	0.68	0.78	0.91	0.84
	Rule Based	82%	0.60	0.89	0.60	0.72	0.81	0.96	0.88
	Hybrid	83%	0.62	0.84	0.67	0.74	0.83	0.93	0.88
	Sampling	92%*	0.85	0.87	0.93	0.91	**0.96**	**0.92**	**0.94**
Coref.	Maj. vote	82%	0.61	0.87	0.84	0.86	0.73	0.77	0.75
	Sampling	90%†	0.80	0.86	0.98	0.92	**0.97**	**0.81**	**0.88**

Table 2.6: Result of different Animacy Models (Bolded according to when our F_1 measure is higher). MFC stands for "Most Frequent Class", MV stands for "Majority Vote" and the other abbreviations stand for features as indicated in the "Features" section. *Estimated ±2% with 95% confidence. †Estimated ±1% with 95% confidence.

I evaluated my models by measuring accuracy, precision, recall, F_1, and Cohen's kappa (κ) compared to the gold-standard annotations. Table 2.6 shows the results for both classes. For referring expression animacy I varied the features to determine the optimal set. I obtained the best result (F_1 of 0.84) using different combinations of three features: noun (N), grammatical subject (GS) and semantic subject (SS). My hybrid model for referring expression animacy performed better (F_1 of 0.88) than the statistical model (F_1 of 0.84). The rule-based model achieved 0.88 F_1 when I applied the rules first, and marked any remaining referring expressions as majority class. The rule based model performed similarly to the hybrid model, but the hybrid model is more consistent.

For the coreference animacy model, I implemented the majority vote approach to detect animacy of coreference chain using the best output of referring expression model. Majority vote resulted in an overall F_1 of 0.75, around 3% of coreference chains resulted in a tied vote, and these were marked as inanimate (the majority class).

I also evaluated my model using direct sampling [Saunders et al., 2009]. I ran the hybrid model over 200 news articles from the OntoNotes [Hovy et al., 2006] data set containing 46,088 referring expressions and 7,836 coreference chains. I randomly sampled 558 coreference chains and checked their animacy markings by hand, resulting in a estimated accuracy of 90% ±2% at a 95% confidence level, as well as estimated precision, recall, and F_1 listed in Table 2.6. Those coreference chains contained 3,543 referring expressions, which allowed me to estimate the accuracy of the referring expression model at 92% ±1% at a 95% confidence level.

The data contains 46 folktales, which have 142 mentions of 12 characters who are members of traditionally inanimate classes (e.g., stoves that walk, trees that talk). I manually identified those 12 characters and evaluated my model's performance on them. The system is able to detect the animacy of these unusual referents with an F_1 of 0.95. Conversely, there was only one mention of a normally animate class that was inanimate in context ("dead horse"), and this was correctly marked by the system.

2.7 Error Analysis

A detailed error analysis of the results revealed at least three minor problems for the hybrid model : short chains, quotes, and exceptions to the rules.

Short coreference chains

Determining the animacy of short coreference chains is still challenging for my system: approximately 11% of short chains are incorrectly marked. As the length of a chain tends toward a single referring expression, the coreference classifier should converge to the referring expression classifier performance. However, for chains between two and four referring expressions long, the majority voting approach seems to fall short. I believe another approach to solving this problem is to generate new rules in my hybrid model so that it can handle these type of special cases.

Quotes

Many quotes are still full of animate words, approximately 2.5% of quotes that are referring expressions are incorrectly marked, and handling this likely will require rule-based processing.

Exceptions to the rules

Finally, a common error type was exceptions to the rules. In the hybrid system I combined together a large number of similar referring expressions under one rule so that I can handle them under a similar animacy class. But there are always exceptions for every rule: for example, I define “it” as inanimate but of course sometimes “it” can refer to an animate object. For the most part these individual instances will be out-voted by animate referring expressions in long chains, so it is a relatively small problem. One approach to solving this would be to implement the idea of Orăsan and Evans [Orăsan and Evans, 2001, 2007] to use supervised machine learning to mark unseen WordNet senses by their animacy rather using specific rules.

2.8 Confirming Generalizability

My animacy work demonstrated a new approach to detecting animacy where animacy is considered a direct property of coreference chains (and referring expressions) rather than words. I combined hand-built rules and machine learning (ML) to identify the animacy of referring expressions and used majority voting to assign the animacy of coreference chains, and reported high performance of up to 0.90 F_1. I also ran some experiments to verify that the approach generalizes to two different corpora (OntoNotes and the Corpus of English Novels) and I confirmed that the hybrid model performs best, with the rule-based model in second place. My tests apply the animacy classifier to almost twice as much data as my initial study. The results also strongly suggest, as would be expected, the dependence of the models on coreference chain quality. I released our data and code to enable reproducibility.

My initial experiments left some questions as to the generalizability of the detector to other story forms. Here I tested the generalizability of my animacy detector on two new corpora, a news subset of OntoNotes [Weischedel et al., 2013] and the subset of the Corpus of English Novels (CEN) [De Smet, 2008]. I tested all three of the models, specifically, an SVM-based ML, a rule-based model, and a hybrid model combining both. I showed, in agreement with the previous results, that the hybrid model performs best, followed by the rule-based model. The results also suggest that the animacy models have a strong dependence on the quality of coreference chains; in particular, the performance of the models on the CEN data (with automatically computed chains) is much poorer than on OntoNotes and the ProppLearner corpus (with manually corrected chains).

2.8.1 Data

I annotated animacy on two new corpora. First, 94 news texts drawn from the OntoNotes Corpus [Weischedel et al., 2013]. Second, 30 chapters from 30 novels drawn from CEN. I performed this manual annotation by following the same guidelines described in (§2.3). In accordance with the procedure, I have annotated the coreference chains of these two corpora as to whether each coreference chain head acted as an animate being in the text. Because the inter-annotator agreement for this annotation was quite high, I only performed single annotation. Details of the corpora are given in Table 2.7. These corpora contain approximately twice as much data, by count of referring expressions and coreference chains, as the initial work.

OntoNotes (ON)

OntoNotes (ON) [Weischedel et al., 2013] is a large corpus containing a variety of genres, e.g., news, conversational telephone speech, broadcast, talk show transcripts, etc., in English, Chinese, and Arabic. I extracted 94 English broadcast news texts that had coreference chain annotations and annotated the animacy of the coreference chains.

Corpus of English Novels (CEN)

Corpus of English Novels (CEN) [De Smet, 2008] contains 292 English novels written between 1881 and 1922 comprising various genres including drama, romance, fantasy, etc. I selected 30 novels and listed the characters of these novels from the online resources. Then I extracted a single chapter of each novel that contains a significant number of characters. I computed coreference chains using Stanford CoreNLP [Manning et al., 2014], and annotated those chains for animacy.

			Ani.	Inani.			
		Ref.	Ref.	Ref.	Coref.	Ani.	Inani.
Corpus	**Texts**	**Exp.**	**Exp.**	**Exp.**	**Chains**	**Chains**	**Chains**
Previous	142	34,698	22,052	12,646	10,941	3,832	7,109
ON	94	4,197	2,079	2,118	1,145	472	673
CEN	30	70,379	20,937	49,442	17,251	2,808	14,443
Total	124	74,576	23,016	51,560	18,396	3,280	15,116

Table 2.7: Counts of various text types. Ref. Exp. stands for Referring Expression; Coref. stands for Coreference; Ani. stands for Animate; Inani. stands for Inanimate.

2.8.2 Models

My animacy model first classifies the animacy of referring expressions, and second classifies each coreference chain as animate or not by taking the majority vote of it's constituting referring expressions. In these experiments I ran my three referring expression animacy detection models and the single coreference chain animacy detection model (majority vote backed by the different referring expression models, which were determined by to be the best coreference model).

SVM Model

SVM Model is a supervised SVM classifier [Chang and Lin, 2011] for assigning animacy to referring expressions, with a Radial Basis Function Kernel where SVM parameters were set at $\gamma = 1$, $C = 0.5$ and $p = 1$. The features of the best performing model are boolean values of whether a given referring expression contained a noun, a grammatical or a semantic subject. I chose these features because animate references tend to appear as nouns, grammatical subjects, or semantic subjects. When training and testing on the same dataset, I used ten-fold cross validation, and reported the micro-averages across the performance on test folds.

Rule-Based Model

The second approach is a rule-based classifier that marks a referring expression as animate if its last word was: (a) a gendered personal, reflexive, or possessive pronoun (i.e., excluding *it*, *its*, *itself*, etc.); (b) the semantic subject to a verb; (c) a proper noun (i.e., excluding named-entity types of LOCATION, ORGANIZATION, MONEY); or, (d) a descendant of LIVING_BEING in WordNet. If the last word of a referring expression is a descendant of ENTITY but not a descendant of LIVING_BEING in WordNet, the model considers it inanimate.

Hybrid Model

Hybrid Model is the third approach where hand-built rules are applied first, followed by the ML classifier to those referring expressions not covered by the rules.

Majority Vote Model

The coreference model applies majority voting to combine the results of the referring expression animacy model to obtain a coreference animacy prediction. For ties, the chain was marked inanimate.

2.8.3 Experiments

I investigated four training setups for the SVM and Hybrid referring expression models: first, training the model each data set individually, and also training on all three datasets together. For all models (SVM, Hybrid, Rule-Based) I also varied the test corpus. Where the test data was a subset of the training data, I applied ten-fold cross-validation. In all approaches, I used the majority vote classifier to identify the animacy of the coreference chains. These experiments are used to compare the performance of the referring expression animacy model on the new corpora, as well as determine the performance for determining coreference chain animacy.

2.8.4 Findings

The results in Table 2.8 and Table 2.9 showed that the hybrid model outperformed all of the other models in detecting referring expression animacy, which is the same result I got from my previous experiments. It performed the best on the previous data, achieving an F_1 of 0.88, and is the most useful model when applying as input to the majority vote model to identify the animacy of coreference chains, achieving an F_1 of 0.77.

The rule-based model performs second-best. It performed best on the previous data for referring expressions, achieving an F_1 of 0.88. But the majority vote model achieved the best result (F_1 of 0.76) on OntoNotes when the rule-based results are used to detect the chain animacy. I developed a baseline for chain animacy where I considered the first referring expression only instead of majority vote and achieved an F_1 of 0.69 and 0.43 on OntoNotes and CEN.

The SVM model performed worse in most of the cases, especially when the outputs are used for the majority vote model. It performed worst when it trained on the Corpus of English Novels and tested on the previous data, achieving an F_1 of only 0.56 for the referring expressions and achieved an F_1 of 0.37 when the results of the referring expressions are used for the majority vote model.

The majority vote model performed best when tested on OntoNotes. It performed worst when tested on the Corpus of English Novels (CEN). Besides the text genre, the

		SVM		Hybrid		Rule-Based	
Train Corpus	**Test Corpus**	$\mathbf{F_1}$	κ	$\mathbf{F_1}$	κ	$\mathbf{F_1}$	κ
Initial	Initial	*0.84*	*0.53*	*0.90*	*0.70*	*0.88*	*0.60*
Initial	OntoNotes	0.70	0.35	0.80	0.54	-	-
Initial	English Novels	0.75	0.53	0.80	0.60	-	-
OntoNotes	Initial	0.82	0.51	**0.88**	**0.64**	-	-
OntoNotes	OntoNotes	0.70	0.36	0.80	0.54	0.76	0.44
OntoNotes	English Novels	0.76	0.54	0.80	0.61	-	-
English Novels	Initial	0.56	0.22	**0.88**	**0.64**	-	-
English Novels	OntoNotes	0.70	0.37	0.80	0.54	-	-
English Novels	English Novels	0.76	0.55	0.80	0.61	0.75	0.48
All	All	0.80	0.53	0.84	0.62	0.82	0.54

Table 2.8: Performance of the majority vote **referring expression** animacy model backed by different referring expression models for different training and testing setups. κ = Cohen's kappa [Cohen, 1960]. Note that the rule-based model does not require training, and so results are not reported for different training combinations. Italics in the first line are the initial results.

major difference between these corpora is the quality of the coreference chains. For OntoNotes, they are manually corrected, while I automatically computed those on CEN. This strongly suggests that the quality of coreference chains is a major factor in the performance of the animacy classifier.

Finally, the results on the combined corpus are reasonable for the referring expression models but performed poorly for the majority vote coreference chain model. This is perhaps to be expected because CEN is the largest corpus among the three and the coreference chains are poor in quality.

Overall, these results strongly suggest that the features used in my animacy model are generalizable to domains outside the Russian folklore corpus used as long as high quality coreference chains are available.

		SVM		Hybrid		Rule-Based	
Train Corpus	**Test Corpus**	$\mathbf{F_1}$	κ	$\mathbf{F_1}$	κ	$\mathbf{F_1}$	κ
Initial	Initial	*0.46*	*0.03*	*0.75*	*0.61*	*0.72*	*0.51*
Initial	OntoNotes	0.60	0.34	**0.77**	**0.59**	-	-
Initial	English Novels	0.52	0.40	0.54	0.41	-	-
OntoNotes	Initial	0.62	0.44	0.72	0.56	-	-
OntoNotes	OntoNotes	0.60	0.34	**0.77**	**0.59**	0.73	0.48
OntoNotes	English Novels	0.42	0.40	0.54	0.41	-	-
English Novels	Initial	0.37	0.18	0.72	0.56	-	-
English Novels	OntoNotes	0.60	0.34	**0.77**	**0.59**	-	-
English Novels	English Novels	0.54	0.43	0.54	0.41	0.46	0.28
All	All	0.58	0.42	0.60	0.43	0.54	0.33

Table 2.9: Performance of the majority vote **coreference chain** animacy model backed by different referring expression models for different training and testing setups. κ = Cohen's kappa [Cohen, 1960]. Note that the rule-based model does not require training, and so results are not reported for different training combinations. Italics in the first line are the initial results.

CHAPTER 3

CHARACTER IDENTIFICATION

3.1 Motivation

Characters are some of the most central elements of narratives, and the concept of *character* plays an important role in most definitions of narrative. As an example, Monika Fludernik defines a narrative as "a representation of a possible world . . . at whose centre there are *one or several protagonists* of an anthropomorphic nature . . . who (mostly) perform goal-directed actions . . . " [Fludernik, 2009, p.6; emphasis ours]. This definition clearly states that characters are central to stories *per se*. Therefore, it is natural to assume that character identification is an important step in automatic approaches to story understanding.

A number of approaches have been proposed for automatically identifying characters. Some approaches, for example, have sought to solve the character identification task using domain-specific ontologies [Declerck et al., 2012] or reasoning by reference to an existing case base [Valls-Vargas et al., 2014a]. Others have taken supervised machine learning approaches [Calix et al., 2013, Barros et al., 2019], where a classifier is trained over data annotated by people. Some approaches, e.g., examining characters' social networks [Sack, 2013], take character identification for granted, implementing heuristic-driven approaches over named entities or coreference chains that are not examined for their efficacy. Regardless of approach, all prior work of which we are aware has, unfortunately, had a relatively impoverished concept of *character*, at least from a narratological point of view. In particular, a key aspect of any character is that it *contributes to the plot*—characters are not just any animate entity in the narrative—and all prior work essentially ignores this point. Therefore, I proposed to incorporate this narratologically grounded definition of character into automatic character identification.

3.2 Approach

I first defined and operationalized the concept of character, and used that concept to generate annotated data (170 narrative texts drawn from 3 different corpora) with high inter-annotator agreement. Then I demonstrated a supervised machine learning model using seven features that performs well (F_1 of 0.94) on these data. The error analysis reveals several choke points (section §3.8) in the performance of the system, most importantly the quality of the co-reference chains.

3.3 An Operationalized Concept of Character

3.3.1 Core Concept of Character

All prior work that tackles the character identification task is unified by it's lack of a clear, operationalized definition of *character*. So far the work that reports the best performance is by [Valls-Vargas et al., 2014a], where they give examples of different types of characters such as humans, animals (e.g., a talking mouse), anthropomorphic objects (e.g., a magical oven, a talking river), fantastical creatures (e.g., goblins), and folkloristic characters (e.g., the Russian characters *Morozko* and *Baba Yaga*). Despite this relatively comprehensive list of character examples, they did not provide a procedure for reliably distinguishing characters from other animate entities in a narrative.

Consider the following example. Let's assume we have a story about Mary, a little girl, and her dog named Fido. Mary plays with Fido when she feels lonely. Also, Fido helps Mary in her daily chores and brings letters to Mary from the post office. One day Mary and Fido are walking through town observing the local color. They see a crowd gathered around a fruit vendor; an ugly man crosses the path in front of them; and another dog barks at Fido. Many narratologists and lay people would agree that the story has at least two characters, Mary and Fido. Depending on how the story is told, either Mary or Fido may be the protagonist. But what about the other entities mentioned in the story? What about the unnamed man who crosses their path? Is he a character? What about the faceless crowd? Is the crowd itself a character, or perhaps its constituent people? What about the fruit vendor, who is hawking his wares? And what about the barking dog? Where do we draw the line?

I noted these problems in prior work, and proposed a preliminary definition of character grounded in narrative theory that addressed these questions. I began by studying different books and literature reviews on narratology that provided different definitions of character. Helpfully, Seymour Chatman, in his classic book "Story and Discourse: Narrative Structure in Fiction and Film" [1980], collected a number of views on character across multiple narratological traditions. Several of the definitions were complex and would be quite difficult to model computationally. Others were too vague to inform computational approaches. However, my definition provided a reasonable target:

> The view of the Formalists and (some) structuralists resemble Aristotle's in a striking way. They too argue that characters are products of plots, that their status is "functional," that they are, in short, participants or *actants* rather than *personnages*, that it is erroneous to consider them as real beings. Narrative theory, they say, must avoid psychological essences; aspects of character can only be "functions." They wish to analyze only what characters do in a story, not what they are—that is, "are" by some outside psychological or moral measure. Further, they maintain that the "spheres of action" in which a character moves are "comparatively small in number, typical and classable." [Chatman, 1980, p.111]

Here, an *actant* is something that plays (i.e., *acts in*) any of a set of active roles in a narrative, and *plot* denotes the main events of a story. This definition, then, though presented via somewhat obscure narratological terminology, gives a fairly conceptually concise definition of a character: a character is *an animate being that is important to the plot*. By this measure then, we are justified in identifying Mary and Fido as characters, but not the various entities they casually encounter in their stroll through town.

3.3.2 What Makes an Entity Important?

My definition considers animate beings who can contribute to the plot as characters. But this definition leads to another problem, namely, how can we measure the importance of the characters? How much of a contribution is enough to be a character? Unfortunately, narratologists' answers are not especially clear, and indeed very few narratologists have attacked this question directly. As Chatman writes, "It is remarkable how little has been said about the theory of character in literary history and criticism" [1980, p.107]. According to the famous cultural theorist and narratologist, Mieke Bal, it is difficult to explain the ideas of *character* because a character so closely resembles a human being. She writes "... no satisfying, coherent theory of character is available is due to this anthropomorphic aspect. The character is not a human being, but it resembles one." [Bal and Van Boheemen, 2009, p.113]. Despite this, for the purposes of reliable inter-annotator agreement to support training and testing effective computational approaches, it is critical for us to define specific tests by which we can decide if an animate being is a character or not.

Chatman points out the importance of the *functionality* of a character with regard to the plot. I formulated my test by starting with the original theoretical work that led to the development of the theory of functionality and actants, namely Vladimir Propp's *Mor-*

phology of the Folktale [1968]. In that theory, Propp describes the concept of a *function*, which is an action or event that drives the plot forward, and is intimately interwoven with the main characters (i.e., the *dramatis personae*). For example, the Villain of a story may cause harm or injury to some member of the Hero's family: Propp names this plot function *Villainy* and assigns it the symbol A. He defined 31 such functions. Prior work on annotating Propp's morphology has shown that the main characters can be reliably identified [Yarlott and Finlayson, 2016], and so those characters which are directly and unambiguously involved in the forwarding of the plot are generally not difficult to identify. These main characters have numerous mentions, close involvement in the main events, and highly distinctive character traits. What about, however, edge cases—potential minor characters—such as the examples in the Mary and Fido above? Minor characters have many fewer mentions, little involvement with main events, and often no uniquely distinguishing traits.

To illustrate the difficulty consider the following example from Propp's data, namely the story *Vasilisa the Beautiful*, which is found in one of my corpora, the extended ProppLearner corpus [Finlayson, 2017]. In this story, the heroine is Vasilisa, whose mother dies right after giving birth to her. Before dying, the mother gave Vasilisa a doll, and the rest of the story concerns how Vasilisa survives the predations of her stepmother with the help of that doll. There is no doubt that Vasilisa is a main character of this story—she is the Heroine—but there is some question about her birth mother. Does the birth mother count as a character, albeit a minor one? I can apply the test of functionality by asking whether the mother's actions or presence are critical to the progression of plot. In particular, the mother gives Vasilisa a critical magical artifact (the doll, which itself become a major character) without which Vasilisa would have been unable to effect much of the action of the story. Because of the mother's involvement, indirect though it may be, in key events of the plot, I can reasonably consider the birth mother a minor character.

In addition to the extended ProppLearn corpus, I also annotated texts from OntoNotes 5.0 [Weischedel et al., 2013] which presented many interesting edge cases. As an example, OntoNotes contains many short news texts, one consisting only of 13 lines about a day in the life of Bill Clinton just before the U.S. election of 2000. In that article "all Americans" is mentioned: "The day got worse when he urged all Americans to vote on November 2.". It is clear that Bill Clinton is a character of this news article because the whole story is about him, but what about the referent "all Americans"? Do they contribute to the "plot" of the article, such as it is? Do they support the development of the main character? In this case, "all Americans" neither effect any functional action in the plot of the article, nor do they contribute anything necessary to the progression of the plot. Indeed, if the reference to "all Americans" was struck from the text, the plot would remain essentially unchanged. Based on this judgement, I do not consider "all Americans" to be a character, even a minor one.

Based on these examples, I can propose a rule for assessing the importance of an entity: if an animate entity is mentioned numerous times, has clear and close involvement in the main events of the plot, and has highly distinguishing character traits, then it is almost certainly a main character. For other animate entities that are mentioned less often, have more tangential connection to the plot, and perhaps lack distinguishing traits, the key test is whether that entity critically contributes to the plot either by directly participating in a important plot event, or enabling the participation of other characters in the plot. In my annotation, I observed that the difficulty of distinguishing characters from non-characters depends strongly on the length of a story. The shorter the text, the harder it is to identify the characters, primarily because there is much less opportunity for entities to present distinct characteristics and contribute clearly to the development of the plot. As a case in point, identifying characters in the third corpus, the Corpus of English Novels [Weischedel et al., 2013], where the chapters are quite long, was easier than identifying characters in the Propp's folktales, and substantially easier than in the short OntoNotes news texts.

3.3.3 Other Aspects of Characters

With a operationalized definition of character now in hand, one might ask whether characters can be further characterized along different dimensions. For example, Ismail Talib [2010] described a number of different possible dimensions of characters: protagonist vs. antagonist, flat vs. round, static vs. developing, and so forth. Propp described seven different types of *dramatis personae*: Hero, Villain, Princess, Helper, Donor, Dispatcher, and False Hero. While these are interesting directions to explore, in this work I did not seek to categorize entities in any way other than character or not.

3.4 Data and Annotation

I with the help of Rahul Mittal annotated characters on 170 texts across three corpora, one with 46 texts (the extended ProppLearner corpus), the second with 94 texts (a subset of the OntoNotes corpus), and the third with 30 texts (a subset of The Corpus of English Novels). Table 3.1 shows the counts of various items of interest across the data. We manually annotated these corpora as to whether each coreference chain acted as a character in the story. Gold coreference chains were already marked on the ProppLearner corpus and OntoNotes, while the coreference chains were automatically computed for the Corpus of English Novel. According to the definition mentioned above, we marked a chain as a character if it is animate and is important to the plot of the story. First, we read the story and find the events important to the plot, there was no agreement across the annotators what the events important to the plot are. Then we assessed the animate objects directly or indirectly involved those events to determine if they were characters or not. As our

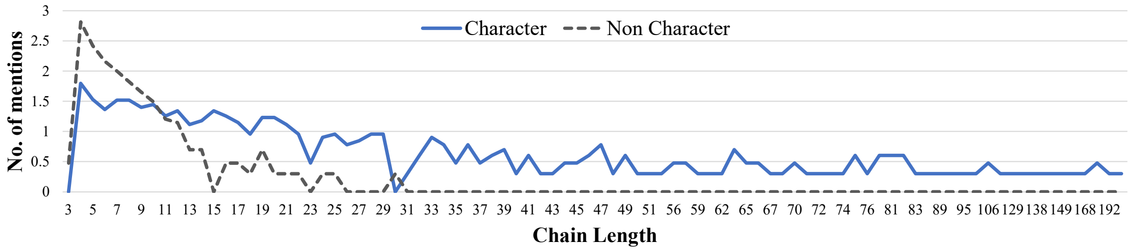


Figure 3.1: Mentions vs. chain: the extended ProppLearner corpus

supervised model is highly dependent on the annotation, therefore, if there are more than one plot, or if the plot is highly subjective, then it should be reflected by the annotators.

Text Types	# Texts	# Coref. Chains	#Ani. Chains	#Inani. Chains	#Char. Chains	#Non-Char. Chains
ProppLearner	46	4,950	2,004	2,946	564	4,386
OntoNotes	94	1,145	472	673	347	798
CEN	30	17,251	2,808	14,443	436	16,815
Total	170	23,346	5,284	18,062	1,347	21,999

Table 3.1: Counts of various text types in the corpus. Coref. stands for coreference; Ani. stands for Animate; Inani. stands for inanimate; Char. stands for character.

3.4.1 The extended ProppLearner (PL)

The extended ProppLearner [Finlayson, 2017] contains gold-standard annotations for referring expressions, coreference chains, and animacy. It comprises 46 Russian folktales originally collected in Russia in the late 1800s but translated into English within the past 70 years.

We double annotated this corpus at the coreference chain level for character, achieving an agreement of 0.78 Cohen's kappa (κ). This level of agreement represents substantial overall agreement [Landis and Koch, 1977]. We discussed any disagreements and corrected them to generate a gold-standard annotation. Our high agreement measures are in accordance with prior work that has shown that *dramatis personae* (i.e., main characters)

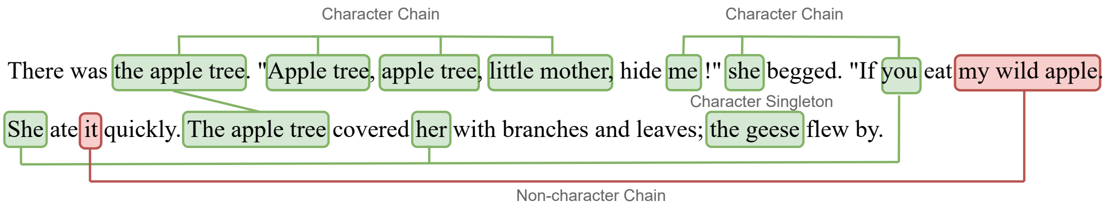

Character Chain
Character Chain
There was the apple tree. "Apple tree, apple tree, little mother, hide me!" she begged. "If you eat my wild apple.
Character Singleton
She ate it quickly. The apple tree covered her with branches and leaves; the geese flew by.
Non-character Chain

Figure 3.2: Sample text fragment of the extended ProppLearner corpus

can be annotated with high reliability. In particular, Yarlott and Finlayson [2016] showed that dramatics personae can be annotated with agreements of $F_1 > 0.8$ and $\kappa > 0.6$. Because of the high agreement for this annotation task, I single-annotated the remaining two corpora for the sake of efficiency.

3.4.2 OntoNotes (ON)

OntoNotes [Weischedel et al., 2013] is a large corpus containing a variety of genres, including news, conversational telephone speech, broadcast news transcripts, talk show transcripts, among others, in English, Chinese, and Arabic. I extracted 94 English broadcast news transcripts that had gold-standard coreference chain annotations and annotated the coreference chains as to character. Despite having clear narrative elements, including characters and events, the news texts have very different goals and textual properties. For example, the plot is only partially represented in a news text, while we have a full plot in many narrative texts.

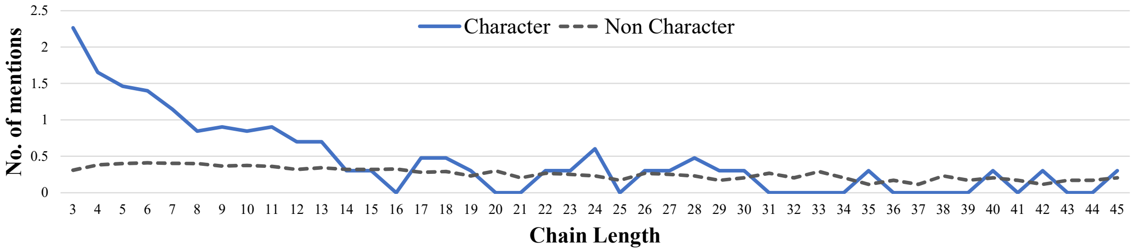


Figure 3.3: Mentions vs. chain: OntoNotes corpus

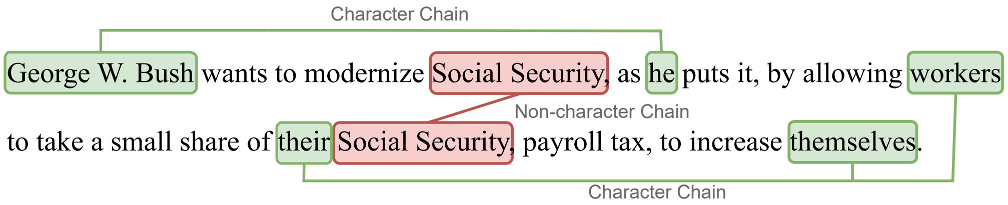

Character Chain
George W. Bush wants to modernize Social Security, as he puts it, by allowing workers
Non-character Chain
to take a small share of their Social Security, payroll tax, to increase themselves.
Character Chain

Figure 3.4: Sample text fragment of the OntoNotes corpus

3.4.3 The Corpus of English Novels (CEN)

The Corpus of English Novels (CEN) [De Smet, 2008] contains 292 English novels written between 1881 and 1922, comprising various genres, including drama, romance, fantasy, adventure, etc. I selected 30 novels and from each extracted a single chapter that contained a significant number of characters. I computed coreference chains using Stanford CoreNLP [Manning et al., 2014], and annotated those chains as to character. I annotated only one chapter per novel due to time constraints; I am aware that in a full novel, the picture might be very different than in a single chapter.

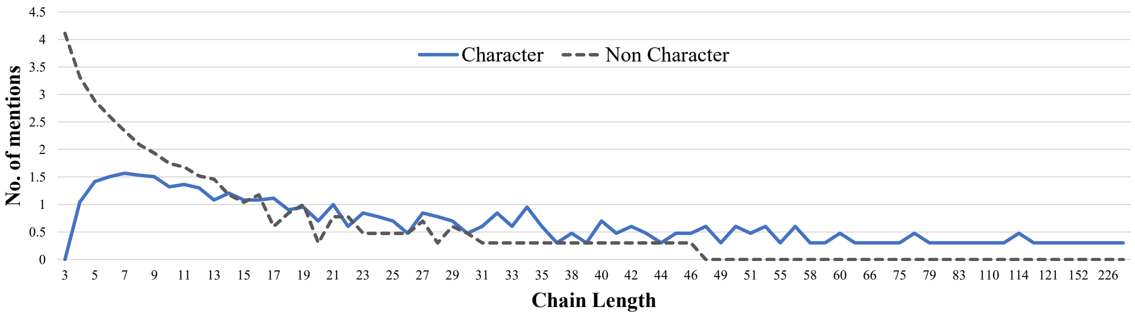
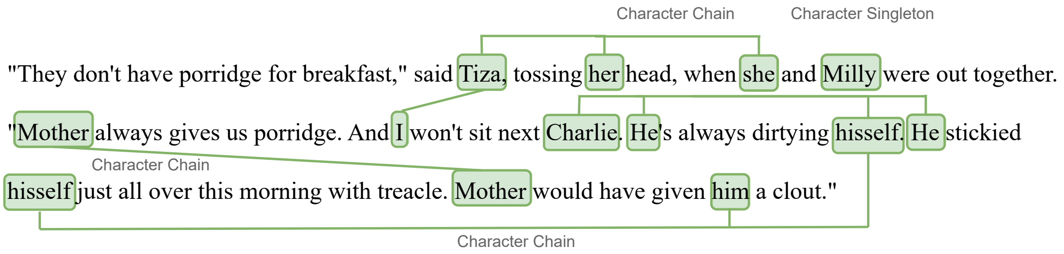


Figure 3.5: Mentions vs. chain: CEN corpus

3.5 Methodology

My character detection model comprises two steps: first, I automatically marked the animacy of coreference chains, and second I applied a supervised machine learning classifier to identify the characters.



"They don't have porridge for breakfast," said Tiza, tossing her head, when she and Milly were out together. "Mother always gives us porridge. And I won't sit next Charlie. He's always dirtying hisself. He stickied hisself just all over this morning with treacle. Mother would have given him a clout."

Figure 3.6: Sample text fragment of the CEN corpus

3.5.1 Animacy Detection

According to my definition of character, it must be an animate object that is important to the plot. Thus one first step to identifying characters is to identify the animate entities. I used my existing animacy classifier for coreference chains, and tried two of their best-performing models, both of which achieved state-of-the-art performance; one is a hybrid model incorporating supervised machine learning and hand-built rules, and the other is a rule-based model consisting of hand-built rules only. As I have gold standard animacy annotation in the extended ProppLearner corpus that allows training the supervised portion of the hybrid model, I trained and ran the hybrid model on this data. For OntoNotes and the Corpus of English Novels, I ran the rule-based model, which did not require gold-standard animacy markings for training, to detect animacy.

3.5.2 Character Classification: Features

I explored seven different integer and binary features to train the character identification model. As I have mentioned earlier, not all animate entities are characters, but all characters are animate entities. Therefore, I incorporated the animacy features while adding additional features for character, and so most of the features are designed to interrogate whether an animate entity acts as a semantic subject of an event or has person-like characteristics. Some of the features are drawn or inspired by prior work.

Coreference Chain Length (CL)

I computed the length of a coreference chain and then normalized the numeric length feature by z score $= (x - \mu)/\sigma$, where x is the raw chain length, μ is the chain length mean, and σ is the chain length standard deviation. This feature explicitly captures the tendency of the long chains to be characters, as discussed in prior work [Eisenberg and Finlayson, 2017].

Semantic Subject (SS)

I also computed whether or not the head of a coreference chain appeared as a semantic subject (ARG0) to a verb, and encoded this as a boolean feature. I used the semantic role labeler associated with the Story Workbench annotation tool [Finlayson, 2008, 2011] to compute semantic roles for all the verbs in the stories. Semantics roles have been previously used for Named Entity Recognition (NER) as seen in [Pang and Fan, 2009]

Named Entity (NE)

I checked whether or not the head of a coreference chain was a named entity with the category *PERSON*, and encoded this as a boolean feature. The named entities were computed using the standard API of the Stanford dependency parse [Manning et al., 2014, v3.7.0].

WordNet (WN)

I detected if the head of a coreference chain is a descendant of *Person* in WordNet, and encoded this as a boolean feature.

Dependency Link (DP)

I computed whether or not the head of a coreference chain appeared as a dependent of `nsubj` dependency link among the enhanced-plus-plus-dependencies of a sentence. The dependencies were extracted using the standard API of the Stanford dependency parse [Manning et al., 2014, v3.7.0] I have used for Named Entity feature. Similar dependencies were used as features elsewhere [Valls-Vargas et al., 2014a].

Triple (TP)

I computed if the head of a coreference chain matches the subject position of any triple and encoded this information as a boolean feature. The triples were extracted from Stanford OpenIE associated with the classic API of the Stanford CoreNLP toolkit [Manning et al., 2014, v3.7.0]. [Goh et al., 2012a] used a similar extraction of an S-V-O triplet.

ConceptNet Feature (CN)

I checked if the head of a coreference chain has any edge that related to *Person* in the ConceptNet semantic network [Speer et al., 2017] and encoded this information as a boolean feature. Features extracted from ConceptNet have also been used as features elsewhere [Calix et al., 2013, Valls-Vargas et al., 2014a].

3.5.3 Character Classification: Models

My character classification model is a simple supervised machine learning classifier with the hand-built features identified above. I used the extended ProppLearner corpus to explore different combinations of features and their importance to model performance. The best-performing model uses all seven features. I then trained and tested this model to the OntoNotes and Corpus of English Novels corpora to see how the model works on different kinds of data sets. The implementation of the character model is done by using an SVM [Chang and Lin, 2011] with a Radial Basis Function Kernel.SVM parameters were set at $\gamma = 1$, $C = 0.5$ and $p = 1$. I have demonstrated the results on different corpora in Table 3.5 and Table 3.6. I trained each model using ten-fold cross-validation, and report macro-averages across the performance on the test folds.

3.6 Preliminary Analysis

Armed with this refined definition of character, I proceeded to generate preliminary data that could be used to explore this idea and demonstrate the feasibility of training a supervised machine learning system for this concept of character. I sought to explore how easily computable features, like those used in prior work, could capture this slightly refined concept of character. I began with the fact that characters and other entities are expressed in texts as coreference chains made up of referring expressions [Jurafsky and Martin, 2007]. Thus any labeling of *character* must apply to coreference chains. I generated character annotations on two corpora, one with 46 texts (the extended ProppLearning corpus) and other with 94 texts (a subset of the InScript corpus), for a total of 1,147 characters and 127,680 words. The annotation procedure is described in the (§3.4) section.

3.6.1 Data

The extended ProppLearner

The ProppLearner corpus was constructed for other work on learning plot functions [Finlayson, 2017]. The corpus that was reported in that paper comprised only 15 Russian folktales, but I obtained the extended set of 46 tales from the authors. These tales were originally collected in Russian in the late 1800's but translated into English within the past 70 years. All of the texts in the corpus already had gold-standard annotations for major characters, congruent with our proposed definition. Usefully, the corpus also has gold-standard annotations for referring expressions, coreference chains, and animacy.

			Coreference Chains				
	Texts	**Tokens**	**Total**	**Ani.**	**Inani.**	**Char.**	**Non-Char.**
ProppLearner	46	109,120	4,950	2,004	2,946	1,047	1,361
Inscript (Subset)	94	18,568	615	105	510	94	521
Total	140	127,680	5,565	2,098	3,467	1,141	1,882

Table 3.2: Counts across coreference chains of different categories, as well as texts and tokens. Ani. stands for animate and Inani. stands for inanimate.

InScript

I also investigated the InScript corpus [Modi et al., 2017]. InScript contains 1,000 stories comprising approximately 200,000 words, where each story describes some stereotypical human activity such as going to a restaurant or visiting a doctor. I selected a subset (94 stories, approximately 19k tokens) of the corpus that describes activity of taking a bath. It has referring expressions and coreference chains already annotated.

3.6.2 Features

I used four different features for our character identification model.

Coreference Chain Length (CL)

I computed the length of a coreference chain as an integer feature. This feature explicitly captures the tendency of the long chains to be characters, as discussed in prior work [Eisenberg and Finlayson, 2017].

Semantic Subject (SS)

I also computed whether or not the head of a coreference chain appeared as a semantic subject (ARG0) to a verb, and encoded this as a boolean feature. I used the semantic role labeler associated with the Story Workbench annotation tool [Finlayson, 2008, 2011] to compute semantic roles for all the verbs in the stories.

Named Entity (NE)

I computed whether or not the head of a coreference chain appeared was a named entity with the category *PERSON*, and encoded this as a boolean feature. The named entities were computed using the classic API of the Stanford dependency parse [Manning et al., 2014, v3.7.0].

WordNet (WN)

I checked if the head of a coreference chain is a descendant of *person* in WordNet, and encoded this as a boolean feature.

3.6.3 Models

My classification model is straightforward supervised machine learning, in which I explored different combinations of the features. I implemented my model using an SVM [Chang and Lin, 2011] with a Radial Basis Function Kernel. SVM parameters were set at $\gamma = 1$, $C = 0.5$ and $p = 1$. We tested different combinations of features on the ProppLearner corpus, and their relative performances are shown in Table 3.4. The best performing feature set was using all four features, and I also tested this model on the InScript data. I trained each model using ten-fold cross validation, and report macroaverages across the performance on the test folds.

3.6.4 Findings

The best model, using all four features, achieves an F_1 of 0.81 on the ProppLearner data, and an F_1 of 0.99 on the InScript data. The result on the InScript data is misleadingly high and deserves some discussion. The InScript stories are quite simple, only told in the first person, and usually featuring only a single animate referent who is also the protagonist. Therefore the almost exclusive reference to characters in these stories was the personal pronoun *I*. Thus both the animacy detector and the character identifier had much higher performance than one would expect on more complicated stories.

		Inanimate				**Animate**			
Corpus	**Acc.**	κ	**Prec.**	**Rec.**	$\mathbf{F_1}$	κ	**Prec.**	**Rec.**	$\mathbf{F_1}$
ProppLearner	85%	0.72	0.93	0.82	0.87	0.72	0.78	0.92	0.84
InScript	99%	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

Table 3.3: Performance of the animacy model on the corpora.

				Non Character			**Character**		
Corpus	**Feature Set**	**Acc.**	κ	**Prec.**	**Rec.**	$\mathbf{F_1}$	**Prec.**	**Rec.**	$\mathbf{F_1}$
Propp-Learner	Baseline MFC	56%	0.0	0.57	1.0	0.72	0.0	0.0	0.0
	SS, WN, NE	80%	0.82	1.0	0.87	0.93	0.75	0.80	0.77
	WN, CL	80%	0.82	1.0	0.87	0.92	0.75	0.80	0.78
	CL, SS, WN	84%	0.78	1.0	0.84	0.92	0.75	0.84	0.79
	CL, WN, NE	82%	0.81	0.86	0.92	0.92	0.82	0.77	0.80
	CL, SS, WN	84%	0.78	1.0	0.84	0.92	0.75	0.84	0.79
	CL, SS, WN, NE	85%	0.78	1.0	0.85	0.91	0.88	0.76	**0.81**
InScript	CL, SS, WN, NE	99%	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**

Table 3.4: Preliminary results of different features sets for identifying characters. MFC stands for most frequent class; Acc. stands for accuracy; Prec. stands for precision; Rec. stands for recall. κ = Cohen's kappa [Cohen, 1960]

3.6.5 Future Direction

A detailed error analysis of the results on the ProppLearner data revealed at least three major problems for the character identification model.

Animacy model

The character model relied on the output of the animacy model, and so if a character was not marked animate, the character model also missed it. Conversely, sometimes inanimate chains are incorrectly marked animate, providing an additional opportunity for the character model to err. Thus, in order to improve the performance of our character model, we have to improve the performance of the animacy model.

Short coreference chains

It is hard to detect a character chain with a very few mentions. To solve this problem we could possibly add some new features related to events of the story because event patterns can be helpful to find a character.

Correlation between animacy and character

Some non-character animate entities were incorrectly identified as characters, because there is strong correlation between animacy and character. To solve this problem we need more analysis of the plot structure and to find features that more specific to character vis-a-vis animacy.

Encoding aspects of the plot

The last point is critical. Although it seems that features related to how animate and prevalent a referent is are quite useful for identifying characters, they still fall somewhat short. I hypothesize that features related to encoding aspects of the plot, to determine if a referent is contributing to the plot in a meaningful way, will be critical to substantially improving character identification performance. I plan to explore this idea in final work.

3.7 Results and Discussion

The extended ProppLearner (PL)

I performed some preprocessing on this corpus, primarily involved in correcting minor errors in the coreference chain annotation. This included removing duplicate coreference chains generated by Stanford CoreNLP, merging coreference chains with the same chain heads, and merging pronouns with the correct chain heads. As expected, I obtained good results using this corpus as the coreference chains are of high quality (i.e., I started with gold standard chains and corrected the small number of errors I found). Table 3.5 and Table 3.6 showed the full set of experiments with the model on each corpus. For the ProppLearner corpus I experimented with different combinations of features as shown. Using all seven features my model achieved an F_1 of 0.81 on this corpus. As mentioned above, I used ten-fold cross-validation. Furthermore, to evaluate the effect on the performance due to the character class imbalances in the animate chains from the animacy classifier, I experimented with two types of class balancing approaches: (1) oversampling the minority class only, and (2) oversampling the minority class and undersampling the majority class. In case 1, the performance improved marginally to an F_1 of 0.84, and in case 2, the performance improved to an F_1 of 0.88.

OntoNotes (ON)

I evaluated the model in three ways on OntoNotes data. First, I trained and tested the character model on the complete OntoNotes data with all features, achieving an F_1 of

				Non Character		
Corpus	**Feature Set**	**Acc.**	κ	**Prec.**	**Rec.**	$\mathbf{F_1}$
Propp-Learner	Baseline MFC	70%	0.0	0.70	0.1	0.82
	CL, WN	86%	0.67	0.89	0.92	0.91
	CL, SS, WN, NE	89%	0.74	0.90	0.97	0.93
	CL, SS, WN, NE, DP, CN	90%	0.76	0.91	0.96	0.93
	CL, SS, WN, NE, DP, TP, CN	89%	0.74	0.91	0.95	0.93
	Over Sampling	84%	0.68	0.82	0.88	0.85
	Over and Under Sampling	88%	0.76	0.87	0.90	0.88
OntoNotes	Baseline MFC	60%	0.0	0.60	0.1	0.74
	CL, SS, WN, NE, DP, TP, CN	49%	0.0	0.0	0.0	0.0
	*Evaluation by Random Sampling**	70%	0.0*	0.0*	0.0*	0.0*
	Over and Under Sampling	24%	0.0	0.18	1.0	0.29
	Over Sampling	87%	0.50	0.76	0.50	0.58
CEN	Baseline MFC	95%	0.0	0.95	0.1	0.97
	CL, SS, WN, NE, DP, TP, CN	97%	0.71	0.97	0.99	0.98
	Over Sampling	94%	0.80	0.95	0.98	0.97
	Over and Under Sampling	90%	0.80	0.88	0.92	0.90
	*Evaluation by Random Sampling**	96%	1.0*	0.96	1.0*	0.98
Weighted Average (by # of Coref Chains)		89%	0.92	0.93	0.95	0.94

Table 3.5: Performance of different features sets for identifying characters. MFC stands for most frequent class; Acc. stands for accuracy; Prec. stands for precision; Rec. stands for recall. κ = Cohen's kappa [Cohen, 1960].

0.66. Because the OntoNotes coreference chains are not completely clean (containing some duplicates and incorrect chains), I used direct sampling [Saunders et al., 2009] to select a subset of the chains and manually corrected them, and trained and tested the full model over this subset. This achieved an improved F_1 of 0.82, suggesting that the classes are imbalanced because the model voted for majority class only. The details of the sampling are: confidence Level = 95% , Confidence Interval = 4, Population = 1,145 and Sample Size = 394. Finally, as the classes are imbalanced (there are many fewer character chains with non-character chains), I performed over- and under-sampling in the same fashion as for the ProppLearner data. When oversampling only, I achieved an improved performance of 0.91 F_1. When over- and under-sampling simultaneously, I achieved a performance of 0.92 F_1.

Corpus	Feature Set	Acc.	κ	Character Prec.	Rec.	$\mathbf{F_1}$
Propp-Learner	Baseline MFC	70%	0.0	0.0	0.0	0.0
	CL, WN	86%	0.67	0.81	0.73	0.77
	CL, SS, WN, NE	89%	0.74	0.91	0.73	0.81
	CL, SS, WN, NE, DP, CN	90%	0.76	0.90	0.77	0.81
	CL, SS, WN, NE, DP, TP, CN	89%	0.74	0.74	0.86	0.81
	Over Sampling	84%	0.68	0.86	0.82	0.84
	Over and Under Sampling	88%	0.76	0.90	0.86	**0.88**
OntoNotes	Baseline MFC	60%	0.0	0.0	0.0	0.0
	CL, SS, WN, NE, DP, TP, CN	49%	0.0	0.50	1.0	0.66
	*Evaluation by Random Sampling**	70%	0.0*	0.50	1.0*	0.82
	Over and Under Sampling	24%	0.0	0.83	1.0	0.91
	Over Sampling	87%	0.50	0.90	0.95	**0.92**
CEN	Baseline MFC	95%	0.0	0.0	0.0	0.0
	CL, SS, WN, NE, DP, TP, CN	97%	0.71	0.87	0.63	0.73
	Over Sampling	94%	0.80	0.91	0.78	0.83
	Over and Under Sampling	90%	0.80	0.91	0.88	**0.90**
	*Evaluation by Random Sampling**	96%	1.0*	1.0*	1.0*	1.0*
Weighted Average (by # of Coref Chains)		89%	0.92	0.91	0.88	**0.90**

Table 3.6: Performance of different features sets for identifying characters. MFC stands for most frequent class; Acc. stands for accuracy; Prec. stands for precision; Rec. stands for recall. κ = Cohen's kappa [Cohen, 1960].

Corpus of English Novels (CEN)

I evaluated our model on CEN in exactly the same way as on OntoNotes. First, I ran our character model on the whole CEN data and achieved an F_1 of 0.73. I used direct sampling to select and correct coreference chains, and the model achieved an F_1 of 1.0 over this corrected data, suggesting that coreference chain quality was a significantly larger factor in performance over this data. The details of the sampling are: confidence Level = 95% , Confidence Interval = 4, Population = 17,251 and Sample Size = 580. Finally, tried oversampling alone to achieve a significantly improved F_1 of 0.83. I also tried simultaneous over- and under-sampling to achieve an improved result of 0.90 F_1.

Corpus	Sampling	Settings
PL	Over	Duplicated 532 char. chains
	Over & Under	Duplicated 532 char. chains, removed 266 non-char. chains
ON	Over	Duplicated 347 char. chains
	Over & Under	Duplicated 225 char. chains, removed 225 non-char. chains
CEN	Over	Duplicated 2,927 char. chains
	Over & Under	Duplicated 6,104 char. chains, removed 10,275 non-char. chains

Table 3.7: Different settings for over and under Sampling.

3.8 Error Analysis

A detailed error analysis of the results revealed some minor problems for the character identification model that depend mainly on the external tools I have used and the quality of the data.

Animacy model

The character model uses the output of the animacy detector and so if a character was not marked animate, the character model also missed it. Conversely, sometimes inanimate chains are incorrectly marked animate, providing an additional opportunity for the character model to err. Thus, the character model's performance is bounded by that of the animacy model. This dependency is shown in Figure 3.7, where the character model performed better when I used the human-annotated animacy labels.

Quality of coreference chains

The quality of coreference chains is critical for the character model. We can see from Table 3.5 and 3.6 that in the initial experiments, my model achieved excellent results for the extended ProppLearner (F_1 of 0.81) data because of its clean and hand-corrected coreference chains. On the other hand, the character model achieved a notably lower performance (F_1 of 0.66 and F_1 of 0.73) on the Ontonotes and CEN corpus, primarily because I have used the automatically generated conference chains for CEN corpus produced by Stanford CoreNLP. This was demonstrated by a random sampling evaluation to manually correct sample of CEN data, after which the model achieved a significantly improved F_1 of 1.0. We need better systems for automatically generating coreference chains to solve this problem.

Short coreference chains

Identifying the character information of short chains is a challenging task because chain length is one of the most effective features of the character model. In the case of short chains, the model only depends on the chain heads, and if a chain head does not carry much meaningful information, then the model can classify that chain incorrectly. We can see the performance improvement of the character model with increasing chain length from Figure 3.7. Solving this problem is critical, but adding more features that carry semantic information of a chain could be helpful.

Imbalanced data

The data should be balanced to obtain good performance, which is a common requirement for any machine learning model. The performance improvement is shown in Table 3.5 and 3.6 for the three datasets after applying under- and over-sampling. The character model achieved the best performance when I applied over- and under-sampling together to ProppLearner (F_1 of 0.88). For OntoNotes, my model reached the best performance when oversampling is applied (F_1 of 0.92). Similarly, the model's performance significantly improved when over and under sampling are applied together to CEN (F_1 of 0.90).

Limited foreign words

One minor source of error for my character model is limited foreign words, which is a data specific problem. The extended ProppLearner data contains numerous Russian character names (e.g., Parakha, Gornya, Shabarsha, etc.) that are not commonly found in English training data for NER systems or linguistic resources (WordNet, ConceptNet). As a result, our system was sometimes not able to identify these chains as a person, and that affects the model's performance. To address this problem, we could, for example, improve coverage of the NER gazetteers.

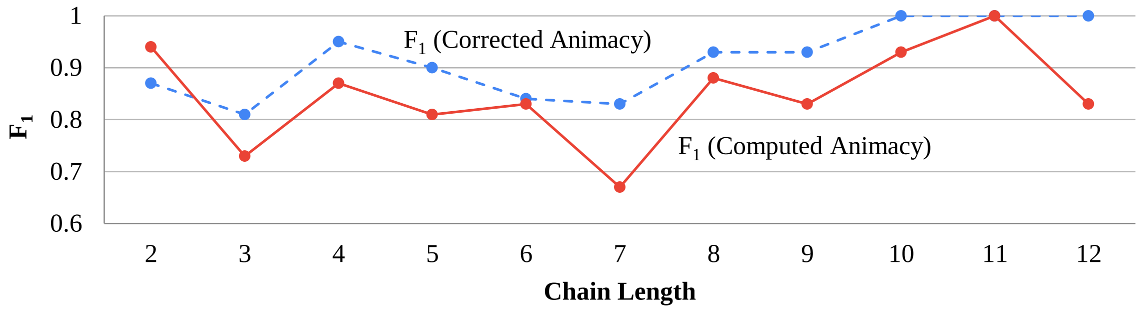


Figure 3.7: F_1 vs. chain length of the character identification model on OntoNotes for both manually corrected and automatically computer animacy markings.

3.9 Confirming Generalizability

I evaluated the generalizability of my model by experimenting with different corpora in training and testing. Table 3.8 shows that the model trained on ProppLearner performed best on every test corpus, and the model trained on OntoNotes performed poorly on others.

The overall performance for these experiments is not as high as the experiments keeping the training and testing corpus the same. As I have discussed before, the three corpora are different in size, type, and structure. The ProppLearner is a well-structured corpus including Russian folktales between 647 and 5,699 words; OntoNotes is a corpus full of

	Test Corpus									**Micro-**
	ProppLearner			**OntoNotes**			**CEN**			**Avg**
Train Corpus	**Acc.**	κ	$\mathbf{F_1}$	**Acc.**	κ	$\mathbf{F_1}$	**Acc.**	κ	$\mathbf{F_1}$	$\mathbf{F_1}$
CEN	84%	0.68	0.84	48%	0.37	0.60	94%	0.85	0.96	**0.93**
ON, CEN	85%	0.72	0.85	50%	0.47	0.62	91%	0.83	0.92	0.89
PL, CEN	86%	0.72	0.86	47%	0.33	0.59	91%	0.83	0.91	0.89
All	85%	0.72	0.85	54%	0.07	0.66	91%	0.83	0.92	0.89
PL, ON	87%	0.74	0.87	81%	0.46	0.88	87%	0.75	0.88	0.88
PL	89%	0.79	0.89	69%	0.27	0.80	87%	0.75	0.88	0.85
ON	71%	0.43	0.76	88%	0.47	0.93	71%	0.42	0.76	0.77

Table 3.8: Performance of **character** model for different training and testing setups. Acc. stands for accuracy. κ = Cohen's kappa [Cohen, 1960].

short broadcast news texts (<1,028 words) that are loosely-structured story-wise; while the CEN corpus includes large chapters from English novels (1,402 - 7,060 words each) where the plot and characters are well developed. As a result, when I run the experiments on different training and testing corpus, the model sometimes finds it challenging to identify the right pattern for another type of corpus.

CHAPTER 4

STEREOTYPICAL ROLE LEARNING

4.1 Motivation

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 demonstrated 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. I present just such an approach here, a k-means-based unsupervised clustering using plot functions as the key feature: I show that if we know characters' involvement in plot functions for a corpus, we can automatically induce the stereotypical roles with reasonable performance.

4.2 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, including 31 basic structural elements or plot functions typical of all fairy tales Russian folklore.

4.2.1 Stereotypical Character Roles

According to Propp in [Propp, 1968], all of the characters in stories can be reduced to the following seven stereotypical character roles.

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.

4.2.2 Plot Functions

Propp described 31 consecutive plot functions in [Propp, 1968].

Absentation One of the members of a family absents himself from home.

Interdiction An interdiction is addressed to the hero.

Violation of interdiction The interdiction is violated.

Reconnaissance The villain makes an attempt at reconnaissance.

Delivery The villain receives information about his victim.

Trickery The villain attempts to deceive his victim in order to take possession of him or his belongings.

Complicity Victim submits to deception and thereby unwittingly helps his enemy.

villainy The villain causes harm or injury to a member of a family.

Lack A member of a family lacks something or desires to have something.

Meditation Misfortune or lack is made known; the hero is approached with a request or command; he is allowed to go, or he is dispatched.

Beginning counteraction The hero agrees to or decides upon counteraction.

Departure The hero leaves home.

First function of the Donor The hero is tested, interrogated, attacked, etc., which prepares the way for his receiving either a magical agent or a helper.

The hero's reaction The hero reacts to the actions of the future Donor.

Provision of a magical agent The hero acquires the use of a magical agent.

Guidance Hero is led to the whereabouts of an object of search.

Struggle The hero and the villain join in direct combat.

Branding The hero is branded.

Victory The villain is defeated.

Liquidation of Lack The initial misfortune or lack is liquidated.

Return The hero returns.

Pursuit The hero is pursued.

Rescue Rescue of the hero from pursuit.

Unrecognized arrival Unrecognized, he arrives home or in another country.

Unfounded claims A false hero presents unfounded claims

Difficult task A difficult task is proposed to the hero.

Solution The task is resolved.

Recognised The hero is recognised.

Exposure The false hero or villain is exposed.

Transfiguration The hero is given a new appearance

Punishment The villain is punished.

Wedding The hero is married and ascends the throne.

4.3 Approach

I implemented K-Means clustering using plot function and thematic information. My approach aims to prove that if we know a character's involvement in the plot functions, we can automatically learn its role.

4.4 Data and Annotation

I demonstrate my method on the so-called *extended* ProppLearner corpus [Jahan et al., 2020a], 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. To the best of my 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, 2020a]. I along with my mentee, Rahul Mittal performed some manual correction on this corpus, primarily elim-

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

Table 4.1: Counts of different archetypes of the gold-standard annotation and the automated output of the animacy-character-archetype model.

inating minor errors in the coreference chain and plot function annotation and merging coreference chains that were erroneously split. Table 4.1 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.

4.5 Methodology

My approach assumes I begin with coreference chain annotations. I first detected the animate entities using my existing state-of-the-art animacy detector [Jahan et al., 2018], then identified which of those animate entities are characters using my existing character identifier [Jahan et al., 2020a]. Finally, I implemented k-means clustering to learn the stereotypical roles of those characters.

4.5.1 Animacy Detection

According to the operational definition of character found in Jahan et al. [2020a], 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. I used my animacy classifier described in the chapter (§2) for animacy detection over coreference chains. I used the best-performing model, a hybrid model incorporating supervised machine learning and hand-built rules.

4.5.2 Character Identification

For identifying characters, I used the character identifier and the gold-standard character annotation of the character identifier I described in the chapter (§3). The character model is a straightforward supervised machine learning model that includes seven features, and it performs quite well on the extended ProppLearner corpus.

4.5.3 Clustering: Models

To cluster identified characters into Propp's stereotypical character role groups, I used k-means clustering. Although Propp identifies seven roles, I excluded the *False Hero* characters from the data because there are only two examples. I have added an extra label named *Others* which represents non-archetype characters or non-major characters.

4.5.4 Clustering: Features

tf-idf

I 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. Using *sci-kit learn* the *tf-idf* parameters were *max_df* = 0.1, *min_df* = 0.01, and *stop-words* = "english".

Bag-of-words

I 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. The parameters of the bag-of-words are configured as *max_df* = 0.1, *min_df*= 0.01, and *stop-words* = “english”.

Hashing

I calculated hashing vectors to convert the words of the coreference chain heads to a sparse matrix of token occurrence counts. The parameters of the bag-of-words are configured as *n_features* = 52.

Plot

I 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. I 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. I 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.

4.5.5 Clustering: Cluster Evaluation Method

Because the output of the k-means clustering is just a set of clusters, to evaluate against the gold standard I must assign a stereotypical character role to each cluster. To do so, I followed the following procedure: (a) Order the list of seven stereotypical character role labels by their gold-standard annotation 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. I explored variations of this procedure using counts and percentages instead of F_1, but the final result was unchanged.

4.6 Results and Discussion

For each feature set explored, I 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. My 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. I 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.

Features	**Hero**	**Villain**	**Donor**	**Helper**	**Princess**	**Dispa-tcher**	**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, T, B, H	**0.60**	**0.63**	0.27	**0.37**	**0.67**	0.40	0.68	**0.29**	**0.58**

Table 4.2: Performance of the different feature sets for $k = 7$. ARI = Adjusted Rand Index, T = *tf-idf*, B = bag-of-words, H = Hashing

4.7 Error Analysis

A detailed error analysis of the results revealed some minor problems for the archetype model that depend mainly on the external tools I have used and the quality of the data.

Animacy and character models

The archetype model uses the output of the animacy and character detectors. So if a character was not marked animate, the character model also missed it, and therefore, the archetype model missed it. Conversely, sometimes inanimate chains are incorrectly marked as animate or character, providing an additional opportunity for the archetype model to err. Thus, the archetype model's performance is bounded by that of the animacy and character models.

Quality of coreference chains

The quality of coreference chains is critical for the archetype model. Initially, my archetype model did not achieve good performance, but the performance increased gradually when some hand-corrections of the coreference chains were done. Besides, there are some mentions where multiple characters are present in plural forms in the coreference chains. Thus, my model sometimes can not perform the string match successfully. Manual correction can be done to solve this problem.

Partial involvement of the characters

Some characters are partially involved in the plot functions; therefore, the archetype clustering model can not successfully cluster them in the right cluster. Exclusion of these data points can be a quick solution, but I did not exclude them from the corpus due to a lack of data points.

Multiple roles

A few characters have multiple roles simultaneously, but my model can learn only one role for a specific character. To resolve this issue, one might want to implement hierarchical clustering on other approaches that support multiple roles simultaneously. I did not implement these approaches because most of the characters in extended proppLearner have only one role.

Plot function annotation

The plot function is an important part of this stereotypical role learning project. Initially, I used the gold annotations that were already present in the extended ProppLearner corpus. Later, I revised the annotation, and the model achieved better performance with the revised annotations.

CHAPTER 5

RELATED WORK

5.1 Animacy Detection

I divided the related work for animacy into two sections: first animacy detection in English, followed by animacy detection in other languages. The work reported here is in English (thus the related work of the first section), but the material covered in non-English second section makes clear both that my approach had not attempt before in any language, and also that no language-specific features have been used in any prior work. There have been both rule-based and machine learning methods to classify the animacy of words, but to the best of my knowledge, no one has combined both techniques, and no one has tackled animacy classification at the referring expression or coreference level.

5.1.1 Animacy Detection in English

Evans and Orăsan [2000] performed animacy classification to improve anaphora resolution using a rule-based method to identify animate WordNet hypernym branches. In later work they used supervised machine learning to mark unseen WordNet senses for their animacy [Orăsan and Evans, 2001, 2007]. The rule-based method uses the unique beginners in WordNet for classification of sense animacy using a statistical chi-squared method, while the machine learning method uses k-nearest neighbors in a multi-step procedure, along with careful feature engineering, to determine noun animacy. They achieved an F_1 of 0.94 for animacy, and also performed an extrinsic evaluation using the MARS anaphora resolution system and a word sense disambiguation algorithm. Similarly, Moore et al. [2013] combined a majority vote model using rule-based methods, features from Word-

Net, and a SVM to achieve an accuracy of 89% for majority voting and 95% for SVM (no F_1 score was reported).

Bowman and Chopra [2012] used a maximum entropy classifier to predict multiple classes for noun phrases as *human*, *vehicle*, *time*, *animal*, etc., with an overall accuracy of 85%. A binary animacy classification could be derived from each of these classes, with a performance of 94% accuracy.

Additionally, there are others that have used pure rule-based and pattern matching methods. Ji and Lin [2009] generate n-grams and performed pattern matching using the Google n-gram corpus to label gender and animacy properties for words for to assist in person mention detection. With these gender and animacy markings, they applied a confidence estimation which is compared against the test document using fuzzy matching. The highest F_1 they achieved for animacy was 0.67, with an F_1 of 0.46 for gender.

Declerck et al. [2012] used an ontology-based method to detect characters in folktales. Their ontology consists of family relations as well as elements of folktales such as supernatural entities. After looking at the heads of noun phrases and comparing them with labels in the ontology, they added the noun phrase to the ontology as a potential character if a match was found. Then, they applied inference rules to the candidate characters in order to find two strings in the text that refer to the same character. They discarded strings that are related to a potential character only once and are not involved in an action. They obtained an accuracy of 79%, a precision of 0.88, a recall 0.73, and an F_1 of 0.80.

Wiseman et al. [2015] used a mention-ranking approach for coreference resolution, using animacy as a feature, derived from the Stanford Coreference System [Lee et al., 2013]. The Stanford Coreference System set animacy attributes using a static list for pronouns, named entity labels, and a dictionary.

Finally, a marginally related rule-based system was implemented by Goh et al. [2012a] using verbs and WordNet in order to determine the protagonists in fairy tales (where protagonists must of necessity be animate). They used the Stanford parser's phrase structure trees to obtain the subjects and objects of the verbs and used the dependency structure to obtain the head noun of compound phrases. Additionally, they used WordNet's *derivationally_related* relation to find verb associated with a particular nominal action. They achieved a precision of 0.69, a recall of 0.75, and an F_1 of 0.67.

5.1.2 Animacy Detection in Other Languages

Nøklestad [2009] implemented animacy detection for Norwegian nouns, using this along with Named Entity Recognition to improve the performance of anaphora resolution. They explored various pattern matching methods, using web data to extract lists of animate nouns as well as to check the animacy of a particular noun. For example, if a noun co-referred frequently with *han* (he) or *hun* (she), then it was characterized as animate. This method achieved an accuracy of 93%. The main problem here, from my point of view, is that using data from the web makes the problem too general: you only measure the typicality of animacy, not the animacy of an item in context. In the case of folktales, we have unusual animate entities (e.g., talking stoves) that will on the whole be seen by the web as inanimate.

Bloem and Bouma [2013] developed an automatic animacy classifier for Dutch nouns by dividing them into *Human*, *Nonhuman* and *Inanimate* classes. They use the k-nearest neighbor algorithm with distributional lexical features—e.g., how frequently the noun occurs as a subject of the verb "to think" in a corpus—to decide whether the noun was predominantly animate. Prediction of the *Human* category achieved 87% accuracy, and the large inanimate class was predicted correctly 98% of the time. But, again, this work focuses on individual noun phrases, not coreference chains, and is concerned with the default animacy of the expression, not its animacy in context.

Another implementation of word-level animacy for Dutch was performed by Karsdorp et al. [2015] on folktale texts. Because this work was the highest performing word-level system, many of our features were inspired by their approach. They used lexical features (word forms and lemmas), syntactic features (dependency parses to check which word is a subject or an object), part of speech tags, and semantic features (word embedding using a skip-gram model to vectorize each word). They implemented a Maximum Entropy Classifier to classify words according to their animacy and obtained a good result of 0.93 F_1 for the animate class, by just using the words, parts of speech, and embedding features.

Baker and Brew [2010] performed animacy classification on a multilingual dataset containing English and Japanese. They used Bayesian logistic regression with morphological features, WordNet semantic categories, and frequency counts of verb-argument relations. They obtained 95% classification accuracy. In sum, all the prior work has been for word-level animacy (usually nouns, sometimes noun phrases). In contrast, I focused on characterizing the animacy of referring expressions and coreference chains.

5.2 Character Identification

Prior work on automatic character identification has relied heavily on statistical techniques and linguistic grammar-based techniques. My work is mainly inspired by Calix et al. [2013] who used a Support Vector Machine (SVM) classifier to detect sentient actors in spoken stories. The model compares four different ML classifiers with 83 features (including knowledge features extracted from ConceptNet) and reports an F_1 of 0.86. It was found that certain speech features enhanced the results for non-named entities. However, the model focuses on animacy detection rather than character identification.

A similar line of work by Valls-Vargas et al. [2014a] implemented a case-based approach using the *Voz* system. Apart from linguistic features, the most important features were extracted from WordNet and ConceptNet. Although they reported a 93.49% accuracy for a subset of the Proppian Folktales, it does not give a concrete definition of a character. They also proposed a similarity measure (*Continuous Jaccard*) that compares the entities from the text and case-base of the *Voz* system. Valls-Vargas [2015] further incorporated a feedback loop into *Voz*; this iterative approach improves co-reference grouping, but there isn't an improvement in character identification.

The most recent work on character identification took a supervised ML approach to classify nouns as characters using 24 different linguistic features, including capitalization and possession-based on `Freeling` and `JavaRAP` [Barros et al., 2019]. Out of the different classifiers, *ClassificationViaRegression,* achieved an F_1 of 0.84; however, it only worked for nouns and ignored pronouns.

Other approaches have used NER systems and domain-specific gazetteers in addition to other techniques such as graphs and verb analysis. Vala et al. [2015] proposed an eight-stage pipeline for identifying characters by building a graph where each name is represented as a node, and the nodes representing the same character are connected with edges. NER and co-reference resolution are used to populate the graph and connect nodes co-occurring in a chain, respectively. The main heuristics used distinguish between distinct characteristics compares genders (by looking at honorifics) and names. The model achieves an average F_1 of 0.58 on two datasets; however, it is limited to a corpus with characters that can be easily recognized by NER. Goh et al. [2012a] proposed a NER-based approach to identify the protagonists in fairy tales using WordNet and verb features. They used the Stanford parser to extract NE candidates, which is then filtered by verb analysis. They reported an F_1 of 0.67. In further work, Goh et al. [2013] identified the dominant character in fables using the VAHA (Verbs Associated with Human Activity) architecture [Goh et al., 2012b] and taking into account quoted speech, achieving an F_1 of 0.76. The same architecture, when applied to news articles, achieves an

F_1 of 0.88 [Goh et al., 2015]. Vani and Antonucci [2019] has described a modular tool called NOVEL2GRAPH, which generates visual summaries of narrative text. As part of the first module, characters are detected using Stanford's NER, which are further filtered using part-of-speech tagging. Character aliases are grouped using the DBSCAN clustering algorithm and stored in a dictionary. They did not report the performance of their approach.

Lastly, Declerck et al. [2012] demonstrated an ontology-based approach for automated character identification in folktales. They compared indefinite noun phrases with ontology labels, and used the matches to propose potential characters. Finally, they applied inference rules, and all occurrences of a particular ontology label were marked as references to the same character. The study reports an F_1 of 0.80. Although this approach has the closest implicit definition of a character to mine, the ontology is domain-based and is unlikely to generalize well to other domains.

5.3 Stereotypical Role Learning

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] in which the authors 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., 2020a], 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.

CHAPTER 6

CONTRIBUTIONS

In my dissertation, I focused on three major chapters that are critical to the field of narrative and language understanding.

6.1 Animacy Detection

In the area of animacy detection, I made five significant contributions. To begin, I reframed the animacy classification problem as one of marking animacy on coreference chains, as opposed to all previous work that attempted to label animacy at the word level. Second, I presented a hybrid framework that combines an SVM classifier with hand-written rules to directly predict the animacy of referring expressions, with an output of 0.90 $F1$, which is comparable to the state of the art for word-level animacy detection. Third, to determine the animacy of coreference chains, I used a majority voting method. In contrast to our preliminary analysis, the overall performance of this method has significantly improved. Fourth, I issued 15 texts with word-level animacy annotations and 142 texts with coreference chain animacy annotations, as well as the code that reproduced the findings. Finally, I tested and confirmed the generilizability of my proposed animacy models. My code and data is publicly available for the community (link: `https://dspace.mit.edu/handle/1721.1/116172`). Additionally, I have two workshop publications ([Jahan et al., 2017, 2020b]) and one conference publication ([Jahan et al., 2018]) on animacy work.

6.2 Character Identification

I made four major contributions in the area of character identification. First, I proposed a more appropriate definition of *character*, contrasting with prior computational works which did not provide a theoretically grounded definition. Additionally, I reported the findings of a review of the literature that is helpful to delineate and define the concept of character. Second, I annotated 170 texts for character, generating data that will be useful for the community. Third, I demonstrated a simple supervised machine learning classifier for character identification that achieved a weighted average of 0.90 $F1$, setting a new benchmark for this task. Finally, I checked and validated my proposed character model's generilizability. My code and data is publicly available for the community. (`https://doi.org/10.34703/gzx1-9v95/RB6ZH0`) I also have one workshop publication ([Jahan and Finlayson, 2019]) and one conference publication ([Jahan et al., 2020a]) on character work.

6.3 Stereotypical Role Learning

In the field of stereotypical role learning, I made two significant contributions. First, I planned and built a pipeline to learn stereotypical roles automatically. All prior work is done with some prior knowledge of stereotypical roles, while my work is done in a completely unsupervised fashion. To the best of my knowledge, this is the first attempt to learn stereotypical roles automatically. Second, I demonstrated the importance of plot functions and thematic role information in clustering similar archetypes. In addition, I will make code and data available to the public.

BIBLIOGRAPHY

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

Norbert Guterman. *Russian Fairy Tales*. Pantheon Books, New York, 1975.

Monika Fludernik. *An Introduction to Narratology*. Routledge, New York, 2009.

Constantin Orăsan and Richard J Evans. NP animacy identification for anaphora resolution. *Journal of Artificial Intelligence Research*, 29(1):79–103, 2007.

Samuel R Bowman and Harshit Chopra. Automatic animacy classification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT): Student Research Workshop*, pages 7–10. Montreal, Canada, 2012.

Folgert Karsdorp, M Meulen, Theo Meder, and APJ van den Bosch. Animacy detection in stories. In *Proceedings of the 6th Workshop on Computational Models of Narrative (CMN'15)*, pages 82–97. Atlanta, GA, 2015.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 2007.

Labiba Jahan, Geeticka Chauhan, and Mark Finlayson. A new approach to animacy detection. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 1–12, Santa Fe, NM, 2018. URL `http://aclweb.org/anthology/C18-1001`. Data and code may be found at `https://dspace.mit.edu/handle/1721.1/116172`.

Graham Alexander Sack. Character Networks for Narrative Generation: Structural Balance Theory and the Emergence of Proto-Narratives. In Mark A Finlayson, Bernhard Fisseni, Benedikt Löwe, and Jan Christoph Meister, editors, *the 4th Workshop on Computational Models of Narrative (CMN'13)*, pages 183–197, Hamburg, Germany, 2013.

Thierry Declerck, Nikolina Koleva, and Hans-Ulrich Krieger. Ontology-based incremental annotation of characters in folktales. In *Proceedings of the 6th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 30–34. Association for Computational Linguistics, 2012.

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

Ricardo A Calix, Leili Javadpout, Mehdi Khazaeli, and Gerald M Knapp. Automatic detection of nominal entities in speech for enriched content search. In *Proceeedings of the 26th International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pages 190–195, St. Pete Beach, FL, 2013.

Labiba Jahan, Rahul Mittal, W. Victor Yarlott, and Mark Finlayson. A straightforward approach to narratologically grounded character identification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6089–6100, Barcelona, Spain (Online), December 2020a. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.536. URL `https://www.aclweb.org/anthology/2020.coling-main.536`.

Harryizman Harun and Zulikha Jamaludin. 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, 2016.

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

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

Labiba Jahan, Geeticka Chauhan, and Mark Finlayson. Building on word animacy to determine coreference chain animacy in cultural narratives. In *The AIIDE-17 Workshop on Intelligent Narrative Technologies WS-17-20*, Salt Lake City, Utah, 2017.

Martha Palmer, Olga Babko-Malaya, and Hoa Trang Dang. Different sense granularities for different applications. In *Proceedings of the 2nd International Workshop on Scalable Natural Language Understanding (ScaNaLU 2004) at HLT-NAACL 2004*, 2004.

Martha Palmer, Hoa Trang Dang, and Christiane Fellbaum. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 13(2):137–163, 2007.

Greg N Carlson and Michael K Tanenhaus. Thematic roles and language comprehension. *Syntax and semantics*, 21:263–288, 1988.

Seppo Kittilä, Katja Västi, and Jussi Ylikoski. Introduction to case, animacy and semantic roles. *Case, animacy and semantic roles*, 99:1–26, 2011.

Michael Connor, Cynthia Fisher, and Dan Roth. Starting from scratch in semantic role labeling: Early indirect supervision. In *Cognitive aspects of computational language acquisition*, pages 257–296. Springer, 2013.

Seppo Kittilä. Object-, animacy-and role-based strategies: A typology of object marking. *Studies in Language. International Journal sponsored by the Foundation "Foundations of Language"*, 30(1):1–32, 2006.

Fernanda Ferreira. Choice of passive voice is affected by verb type and animacy. *Journal of Memory and Language*, 33(6):715–736, 1994.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501. Association for Computational Linguistics, 2010.

Ryu Iida, Kentaro Inui, Hiroya Takamura, and Yuji Matsumoto. Incorporating contextual cues in trainable models for coreference resolution. In *Proceedings of the EACL Workshop on The Computational Treatment of Anaphora*, pages 23–30. Citeseer, 2003.

Claire Cardie and Kiri Wagstaf. Noun phrase coreference as clustering. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. *A Comprehensive Grammar of the English Language*. Longman, London, 1985.

Mutsumi Yamamoto. *Animacy and reference: A cognitive approach to corpus linguistics*. John Benjamins Publishing, Amsterdam, 1999.

Mark A. Finlayson. 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, 2017. doi: 10.1093/llc/fqv067.

Mark A Finlayson, Jeffry R Halverson, and Steven R Corman. The n2 corpus: A semantically annotated collection of islamist extremist stories. In *LREC*, pages 896–902, 2014.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations*, pages 55–60. Baltimore, MD, 2014.

Mark Alan Finlayson. Collecting semantics in the wild: The story workbench. In *Proceedings of the AAAI Fall Symposium on Naturally Inspired Artificial Intelligence*, pages 46–53. Arlington, VA, 2008.

Mark A Finlayson. The Story Workbench: An extensible semi-automatic text annotation tool. In *Proceedings of the 4th Workshop on Intelligent Narrative Technologies (INT4)*, pages 21–24. Stanford, CA, 2011.

Lars Wissler, Mohammed Almashraee, Dagmar Monett Díaz, and Adrian Paschke. The gold standard in corpus annotation. In *IEEE GSC*, 2014.

J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

Christiane Fellbaum, editor. *WordNet An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. http://arxiv.org/abs/1301.3781, 2013.

Deeplearning4j Development Team. Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. `http://deeplearning4j.org`, 2017. Accessed: 2017-04-08.

Lilja Ovrelid. Animacy classification based on morphosyntactic corpus frequencies: some experiments with Norwegian nouns. In *Proceedings of the Workshop on Exploring Syntactically Annotated Corpora*, pages 24–34. Birmingham, England, 2005.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2): 313–330, 1993.

Karin Kipper Schuler. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. PhD thesis, University of Pennsylvania, 2005.

Mark Saunders, Philip Lewis, and Adrian Thornhill. *Research methods for business students*. Prentice Hall,UK, 2009.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60. Association for Computational Linguistics, 2006.

Constantin Orăsan and Richard Evans. Learning to identify animate references. In *Proceedings of the 2001 Workshop on Computational Natural Language Learning (CoNLL)*, page Article No. 16. Toulouse, France, 2001.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. OntoNotes Release 5.0, 2013. LDC Catalog No. LDC2013T19, `https://catalog.ldc.upenn.edu/LDC2013T19`.

Hendrik De Smet. Corpus of English novels, 2008. `https://perswww.kuleuven.be/~u0044428/`.

Cristina Barros, Marta Vicente, and Elena Lloret. Tackling the challenge of computational identification of characters in fictional narratives. In *2019 IEEE International Conference on Cognitive Computing (ICCC)*, pages 122–129, Milan, Italy, 2019.

Seymour Chatman. *Story and Discourse: Narrative Structure in Fiction and Film*. Cornell University Press, Ithaca, NY, 1980. URL `https://books.google.com/books?hl=en&lr=&id=ewrOp9uPjYUC&oi=fnd&pg=PA9&dq=Story+and+Discourse:+Narrative+Structure+in+Fiction+and+Film&ots=p5mhrY2SeN&sig=i36A5anpjy5vBTYWEBcRhAXScPA#v=onepage&q=Story%20and%20Discourse%3A%20Narrative%20Structure%20in%20Fiction%20and%20Film&f=false`.

Mieke Bal and Christine Van Boheemen. *Narratology: Introduction to the theory of narrative*. University of Toronto Press, Toronto, 2009.

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

W Victor H Yarlott and Mark A Finlayson. ProppML: A complete annotation scheme for proppian morphologies. In *7th Workshop on Computational Models of Narrative (CMN 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

Ismail S Talib. Narrative theory: A brief introduction, 2010. Retrieved from `https://courses.nus.edu.sg/course/ellibst/NarrativeTheory/`.

Joshua Eisenberg and Mark Finlayson. A simpler and more generalizable story detector using verb and character features. In *the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2708–2715, Copenhagen, Denmark, 2017.

Wenbo Pang and Xiaozhong Fan. Chinese nominal entity recognition with semantic role labeling. In *2009 International Conference on Wireless Networks and Information Systems*, pages 263–266, Milan, Italy, 2009.

Hui-Ngo Goh, Lay-Ki Soon, and Su-Cheng Haw. Automatic identification of protagonist in fairy tales using verb. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 395–406. Springer, 2012a.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*, page 4444â€“4451, San Francisco, CA, 2017.

Ashutosh Modi, Tatjana Anikina, Simon Ostermann, and Manfred Pinkal. Inscript: Narrative texts annotated with script information. *arXiv preprint arXiv:1703.05260*, 2017.

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

Ruth Anne Robbins. 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, 2005.

Pablo Gervás. Propp's Morphology of the Folk Tale as a Grammar for Generation. In Mark A. Finlayson, Bernhard Fisseni, Benedikt Löwe, and Jan Christoph Meister, editors, *2013 Workshop on Computational Models of Narrative*, volume 32 of *OpenAccess Series in Informatics (OASIcs)*, pages 106–122, Dagstuhl, Germany, 2013.

Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-939897-57-6. doi: 10.4230/OASIcs.CMN.2013.106. URL `http://drops.dagstuhl.de/opus/volltexte/2013/4156`.

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

Jayadev Bhaskaran and Isha Bhallamudi. 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, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3809. URL `https://www.aclweb.org/anthology/W19-3809`.

Richard Evans and Constantin Orăsan. Improving anaphora resolution by identifying animate entities in texts. In *Proceedings of the Discourse Anaphora and Reference Resolution Conference (DAARC2000)*, pages 154–162. Lancaster, England, 2000.

Joshua Moore, Christopher JC Burges, Erin Renshaw, and Wen-tau Yih. Animacy detection with voting models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 55–60, 2013.

Heng Ji and Dekang Lin. Gender and animacy knowledge discovery from web-scale n-grams for unsupervised person mention detection. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 1*, volume 1, 2009.

Sam Joshua Wiseman, Alexander Matthew Rush, Stuart Merrill Shieber, and Jason Weston. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Association for Computational Linguistics and International Joint Conference on Natural Language Processing,*. Association for Computational Linguistics, 2015.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916, 2013.

Anders Nøklestad. *A Machine Learning Approach to Anaphora Resolution Including Named Entity Recognition, PP Attachment Disambiguation, and Animacy Detection*. PhD thesis, University of Oslo, Oslo, Norway, May 2009.

Jelke Bloem and Gosse Bouma. Automatic animacy classification for Dutch. *Computational Linguistics in the Netherlands Journal (CLIN)*, 3:82–102, 2013.

Kirk Baker and Chris Brew. Multilingual animacy classification by sparse logistic regression. *Information Concerning OSDL OHIO STATE DISSERTATIONS IN LINGUISTICS*, page 52, 2010.

Josep Valls-Vargas, Jichen Zhu, and Santiago Ontañón. Narrative hermeneutic circle: Improving character role identification from natural language text via feedback loops. In *Proceedings of the 24th International Conference on Artificial Intelligence*, page 2517â€"2523, Buenos Aires, Argentina, 2015.

Hardik Vala, David Jurgens, Andrew Piper, and Derek Ruths. Mr. bennet, his coachman, and the archbishop walk into a bar but only one of them gets recognized: On the difficulty of detecting characters in literary texts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 769–774, Lisbon, Portugal, 2015.

Hui-Ngo Goh, Lay-Ki Soon, and Su-Cheng Haw. Automatic dominant character identification in fables based on verb analysis â€" empirical study on the impact of anaphora resolution. *Knowledge-Based Systems*, 54:147 – 162, 2013.

Hui-Ngo Goh, Lay-Ki Soon, and Su-Cheng Haw. Vaha: Verbs associate with human activity–a study on fairy tales. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 313–322, Dalian, China, 2012b.

Hui-Ngo Goh, Lay-Ki Soon, and Su-Cheng Haw. Automatic discovery of person-related named-entity in news articles based on verb analysis. *Multimedia Tools and Applications*, 74(8):2587–2610, 2015.

K Vani and Alessandro Antonucci. Novel2graph: Visual summaries of narrative text enhanced by machine learning. In *Text2Story@ ECIR*, pages 29–37, Cologne, Germany, 2019.

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

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

R.H.-G. Chen, C.-C. Chen, and C.-M. Chen. Unsupervised cluster analyses of character networks in fiction: Community structure and centrality. *Knowledge-Based Systems*, 163:800–810, 2019. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2018.10.005. URL https://www.sciencedirect.com/science/article/pii/S095070511830491X.

David Bamman, Brendan O'Connor, and Noah A. Smith. 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, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-1035.

David Bamman, Ted Underwood, and Noah A Smith. 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, 2014.

Labiba Jahan, W Victor H Yarlott, Rahul Mittal, and Mark A Finlayson. Confirming the generalizability of a chain-based animacy detector. *1st Workshop on Artificial Intelligence for Narratives (AI4N 2020)*, 2020b.

Labiba Jahan and Mark Finlayson. Character identification refined: A proposal. In *Proceedings of the First Workshop on Narrative Understanding*, pages 12–18, Minneapolis, MN, 2019.

VITA

LABIBA JAHAN

2010–2014	B.Sc., Computer Science Shahjalal University of Science & Technology Sylhet, Bangladesh
2014–2016	Lecturer Metropolitan University Sylhet, Bangladesh
2016–2019	M.S., Computer Science Florida International University Miami, Florida, USA
2016–2021	Ph.D., Computer Science Florida International University Miami, Florida, USA
2016–2021	Graduate Research Assistant Cognition, Narrative and Cultural Laboratory Florida International University Miami, Florida, USA
2017	Mentor Geeticka Chauhan, Undergraduate Research Assistant Florida International University Miami, Florida, USA
Summer, 2019	Summer Research Intern Product Simulation and Modelling Research Group Siemens Corporate Technology Princeton, New Jersey, USA
2020–2021	Mentor Four Senior Year Project teams Florida International University Miami, Florida, USA
2020–2021	Mentor Rahul Mittal, Undergraduate Research Assistant Florida International University Miami, Florida, USA

PUBLICATIONS

Labiba Jahan, Rahul Mittal, W. Victor H. Yarlott, Mark A. Finlayson (2020). A Straightforward Approach to Narratologically Grounded Character Identification. In Proceedings of the 28th International Conference on Computational Linguistics (COLING), Barcelona, Spain (Online). (pp. 6089-6100)

Labiba Jahan, Geeticka Chauhan, Mark A. Finlayson (2018). A New Approach to Animacy Detection. In Proceedings of the 27th International Conference on Computational Linguistics (COLING), Santa Fe, New Mexico, (pp. 1-12).

Pinchao Liu, Adnan Maruf, Farzana Beente Yusuf, Labiba Jahan, Hailu Xu, Boyuan Guan, Liting Hu, Sitharama S Iyengar (2019). Towards Adaptive Replication for Hot/Cold Blocks in HDFS using MemCached. In Proceedings of the International Conference on Data Intelligence and Security (ICDIS), South Padre Island, Texas, (pp. 188-194).

Labiba Jahan, W. Victor H. Yarlott, Rahul Mittal, Mark A. Finlayson (2020). Confirming the Generalizability of a Chain-Based Animacy Detector. In Proceedings of the 1st Workshop on Artificial Intelligence for Narratives (AI4N), Oklahoma, Japan.

Labiba Jahan and Mark A. Finlayson (2019). Character Identification Refined: A Proposal. In Proceedings of the First Workshop on Narrative Understanding (WNU) co-located with NAACL, Minneapolis, Minnesota, (pp 12-18).

Labiba Jahan, Geeticka Chauhan, Mark A. Finlayson (2017). Building on Word Animacy to Determine Coreference Chain Animacy in Cultural Narratives. In Proceedings of the 10th Workshop on Interactive Narrative Technologies (INT) co-located with AIIDE-17, Salt Lake City, Utah, (pp. 198-203) and in Widening NLP (WiNLP 2018) workshop co-located with NAACL, New Orleans, Louisiana. Non-archival.

Partha Sarathi Kar, Shantanu Mandal, and Labiba Jahan (2016). An Improved Unicode Based Sorting Algorithm for Bengali Words. American Academic & Scholarly Research Journal V(8), No-3. (pp. 9-14)

Labiba Jahan, Umme Kulsum, Abu Naser (2015). Bengali Diphone Duration Modeling for Bengali Text to Speech Synthesis System. American Academic & Scholarly Research Journal, V(7), No-3. (pp. 18-24)