Spring 2022

Heroes, Villains, and the In-Between: A Natural Language Processing Approach to Fairy Tales

Ruby Alling Ostrow
Bard College, ro2159@bard.edu

Heroes, Villains, and the In-Between:
A Natural Language Processing Approach to Fairy Tales

Senior Project Submitted to
The Division of Science, Math, and Computing
of Bard College

by
Ruby Ostrow

Annandale-on-Hudson, New York
May 2022

“In all Chaos there is a cosmos, in all disorder a secret order.”
-CG Jung

To finding the one behind the many.

Acknowledgements

I am incredibly grateful to all of those who have helped me on the arduous journey of completing this project and throughout my time at Bard College in general.

I would above all like to thank my senior project advisor, Sven Anderson, for all of his help this semester. His willingness to engage in detailed discussion of my ideas with me, his wise advice, and skillful guidance through these rather uncharted waters have been indispensable throughout. I would not have been able to create this project without all of his help and support. I am deeply appreciative.

I would also like to thank Keith O'Hara for advising me through the first semester of the senior project process. I am very grateful for his understanding of the struggle of starting one senior project while finishing another and his patience and kindness throughout.

Furthermore, I would like to thank my parents for their endless support of me. They have kindly listened to me prattle on about technical jargon mixed with fairy tale plots ad nauseam and have even made themselves available to me to talk through all aspects, large and small, of my thoughts on the project, whether or not they truly had any interest in the matter at hand. Their presence and support have been utterly necessary and deeply appreciated throughout the creation of this project and throughout my whole time at Bard.

I am also utterly grateful for my dear friends at Bard. They have patiently put up with my bleak periods of stress, kept me company through hours of work, and, when necessary, "dragged" me from my computer with temptations of cheering distractions of goat visits, ice cream breaks, video game adventures, and cowboy-themed poker nights. My gratitude for them and our time together throughout college is eternal.

Finally I would like to thank the Computer Science department as a whole. I have learned so much in my classes here and look forward to taking that knowledge into the world.

Abstract

While great strides have been made with natural language processing (NLP) techniques in the last few decades, there has been a notable lack of research into utilizing NLP for the genre of fiction. This project seeks to address this gap by considering the use of NLP techniques for the summarization of European fairy tales. This subgenre of fiction is an appropriate starting point for investigation due to its archetypal characters and relatively simple story arcs. My approach is to extract the main characters of texts, along with key descriptors in the form of modifying adjectives and verbal actions the characters take part in. Through this method, I suggest how we may parse characters into Proppian archetypes by tracking their probabilistic association with certain linguistic occurrences. This classification schema in turn makes possible the broader classification of fairy tales into types. The model has an overall F1 score of 0.77, the individual parts having F1 scores of 0.89, 0.75, and 0.66 for character retrieval, adjective extraction, and verb extraction, respectively. This project may also be extended further, laying key groundwork for further automatization of categorization of characters and ultimately stories themselves.

Table of Contents

Introduction

What is the essence of a story? Is there an essence to a story? This project is an attempt to apply natural language processing summarization techniques to the works of fiction, particularly the genre of European fairy tales, with an eye to the fundamental features that might compose them. My aim is to determine if it is possible to develop a computational model to extract these defining features from the odd and unpredictable nature of this classic form of literature.

Natural language processing research on fictional texts as a whole is very limited; generally attempts at summarization have focused on genres such as news articles and medical papers. These types of writing follow more formulaic patterns, leading to greater ease of summarization given the knowledge of locations (in sentences or in a document as a whole) with a high probability of key information. But fictional narratives are not so straightforward. Even within a particular genre, the location of crucial details within a text is not set, nor is there a definite form to the sentences in which these details are provided. Indeed, it is not even immediately clear *what* those crucial details may be. There are a variety of what one might define as the "right" ways to summarize a text, whether coming at the question from a literary standpoint or a computational one and without even taking into account the question of genre of text. It is no wonder that NLP has thus far made minimal progress in this area.

But before introducing the approach I have taken to address these basic difficulties we need to ask what our fundamental purpose is. Why might it be important to develop automated summaries of fairy tales? To answer that question we first have to distinguish between different types of summaries. One possibility here would be to attempt to furnish brief synopses of the plots of various stories (the usual task of text summarization efforts in NLP; El-Kassas et al.

2021 and Nazari & Mahdavi 2019 catalog the present state of this research as I discuss briefly below). Perhaps there might be some utility to such an approach, but this is not our concern in this project. Rather, my aim is to provide summaries that give us the basis for overarching *categorizations*–classifications of fairy tales based on shared characteristics. The future ideal of this work would then be to provide potential users–perhaps students or researchers–with a classification schema that will enable them to have a broad overview of the subject matter they are investigating.

The central question is how such a classification schema may be realized. To begin with, we must identify the defining features–the basic, essential elements out of which a narrative is constructed–that would give the means to classify it. In fiction at least these are not immediately identifiable. At first blush, the key events in the development of a tale might appear to constitute the essential elements of which a summary should be made up (Adolfo & Ong 2019, Lovering 2016, and Goyal et al. 2010 are examples of this approach in NLP research into fairytales). However I urge that a better starting point is with the characters who enact the events or are affected by them. These, after all, are the agents which constitute the subject or object of the events that occur; it is these characters who make up the basic landscape of a given story (an approach particularly following the work of W. Zhang et al. 2019). Thus on my approach the first element to be collected is a set of agents (or characters), some of whom are more important than others (Fernandez et al. 2015 is another example of a project concerned with this task).

But obviously a list of characters does not constitute a story or even the outline of one. To further define this summary, we must have a deeper sense of who the key characters are and how, in general, they relate to one another. Here I have found it useful to build on the work of

Vladimir Propp, a linguist who was concerned with identifying central character and story arc archetypes in Russian folktales.[1] He is commonly utilized by others interested in NLP techniques performed on fairytales and specifically in character study, including but not limited to Volkova et al. 2010, Alm & Sproat 2005, El Maarouf et al. 2012, and Declerck et al. 2010.

Propp 1968 establishes the following eight archetypes from his detailed study of 100 folk tales:

1. Hero - protagonist who undergoes quest
2. Princess - character who is generally a love interest or sought by hero
3. Donor - character who gives the hero a key tool for their quest
4. Helper - character who uses knowledge to help hero on their quest
5. Dispatcher - character who sends hero on quest
6. Princess' father - character who rewards the hero
7. False hero - character who takes credit for hero's actions
8. Villain - antagonist to hero

For this project, I have developed a simplification of Propp's archetypes that better fits the Grimm's tales. Firstly, I broaden the princess archetype to 'love interest' as royal princes or princesses equally appear as love interests in Grimm's tales and not necessarily the object of a quest. Secondly, I have broadened the dispatcher archetype to include royalty at large, as the dispatcher in particular is less distinct in Grimm's tales but a royal who takes part in affecting the hero's quest more generally is common. Thirdly, I combine the donor and helper archetypes into a single helper archetype as they do not appear as clearly distinguishable in Grimm's tales; a helper character is often present but both helpful actions and objects are often included in their help. Fourthly, I broaden the archetype of the princess' father to include all non-villainous familial entities, including familial pairs that appear in a subset of stories without a specific hero

[1] While I am using the fairy tales specific to the Brothers Grimm rather than Russian folklore, I follow Volkova et al. 2010, Alm & Sproat 2005, and El Maarouf et al. 2012 in extending his work to folklore more broadly.

character (such as the common husband and wife pair). Finally, the false hero archetype, while perhaps present on occasion, is not so common in Grimm's tales so I will not be considering it.

In this way, we can identify a set of archetypes: heroes (or heroines), love interests, helpers, dispatchers, non-villainous family members, and villains that will be central to fairy tales. This in turn can give us insight into the plot possibilities of a given story. Propp shows how, given folk tale tropes,[2] certain types of actions or events are specific to given character archetypes, such as kidnapping, call for help, and a hero's test (examples that Declerck et al. 2010 furnishes). In this way, with knowledge of the likely archetypes of the main characters the set of possible stories into which a given story might fall is limited.

But how might we extract descriptions of the characters that would give insight into their archetype? I suggest that we may learn about them both from the adjectives that are used to describe them and the verbs they act as subject or object of. This is in keeping with the work of Bamman et al. 2013 in predicting the archetype of characters in film via their descriptors and related verbs. I'll begin with adjectives. Such terms give us a sense of how the characters are viewed by other characters, and also, perhaps more importantly, present the particular author's attitude towards them and the aspect under which they or their actions are to be understood. For example, in *Snow White* (named *Snowdrop* in my dataset) it is critical that the Queen doesn't just perform a series of actions, but that she is "evil." This adjective gives us the lens through which her actions are to be understood.

Fortunately for our purposes in this project, there is in general a clear correlation between certain adjectives and the archetype of the character they describe. To illustrate the importance of

[2] For further study into concretely categorizing stories based on the character archetypes present, further work is needed to extend Propp's character-dependent actions based on Russian folklore to other European fairytale genres in a similar way as I have demonstrated in adapting his character archetype.

adjectives as defining archetype, consider the following figure (Figure 1).[3] Utilizing a subset of the dataset, of size 5 (10% of the total dataset), the figure shows the average probability of an adjective given an archetype.[4]

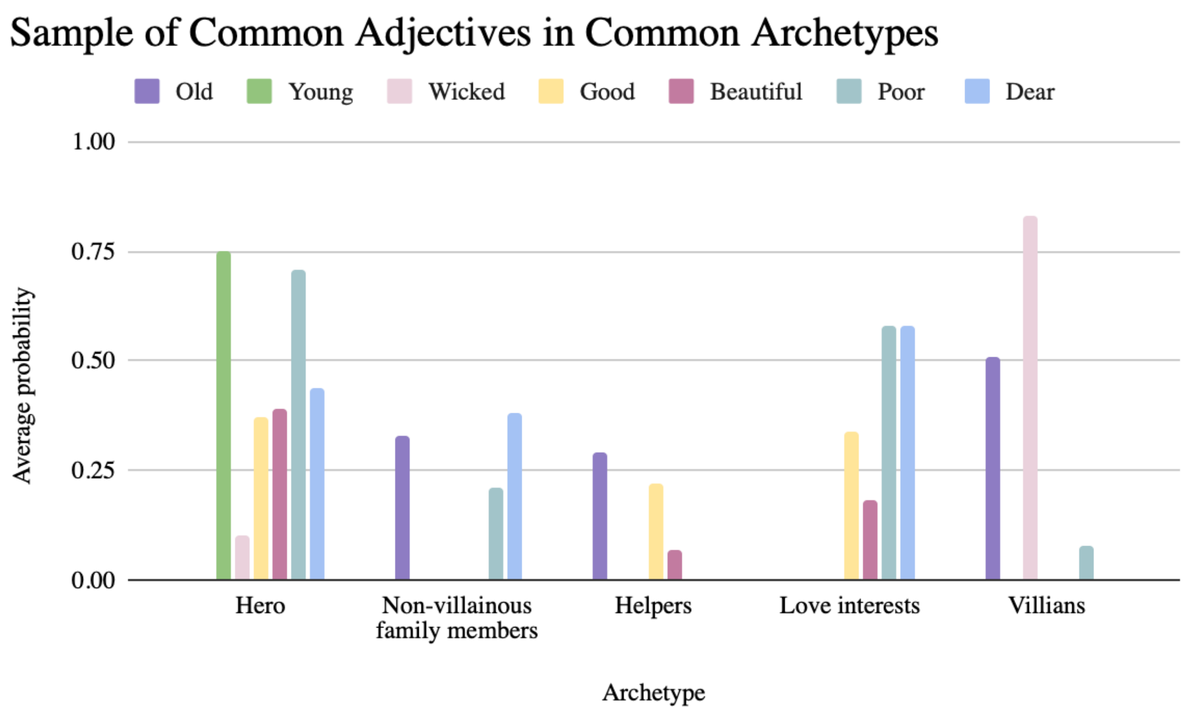

Figure 1: Average probability of each adjective in a given character archetype

While this is certainly not representative of all words that are suggestive of a given archetype (and leaves out the dispatcher given their lack of appearance in the subset of data), it is helpful to see the trends in the usage of certain words across multiple tales. Here we may see that "young" is strongly suggestive of heroes and heroines, in this data subset only occurring with those

[3] The dispatcher archetype is not included as there was not a character fitting this type among the subset. Furthermore, while they follow similar adjective trends as non-villainous family members, they are less clearly distinct by adjectives but are better defined by verbs.

[4] To calculate the probabilities, I annotated each document in the subset for all instances of each word. I then recorded the number of usages in reference to each of the main characters (as representing one of the five archetypes), dividing the number for each archetype by total usages of the word in the document. Finally I averaged the scores determined for each archetype for each word.

characters. “Good,” “beautiful,” “poor,” and “dear” are likewise highly associated with particular archetypes. While some of these words likewise occur in reference to non-villainous family members, helpers, and love interests, we may see that certain groupings of the words are more common from one type to another. In non-villainous family members and helpers, for example, “old” is also a common adjective, which is also true for villains, but these character types may be differentiated by the additional presence of “poor” and “dear” for non-villainous family members and “beautiful” and “good” for the helpers. Thus we see that not only individual adjectives but also certain groupings help us to distinguish the archetype of a given character, showing the utility of the extraction of adjective groups.

But adjectives are not the only way we can sort characters into archetypes. We can also learn of the effect a given character has on the progression of the narrative (or what effect another character has upon this one as to progress the tale). This effect is given through a set of verbs. W. Zhang et al. 2019 focuses simply on adjective character descriptors but I find that, in keeping with Bamman et al. 2013, the addition of verbs is important to character categorization. While I initially hypothesized that only adjectives would be significant, it turned out that verbs were important as well in determining character archetypes, and are often crucial with respect to side characters who are given no adjective descriptors. Verbs of physical action and of feeling (such as running, mounting, and crying) are most common to heroes and heroines, as well as love interests, whereas verbs of speaking have a higher frequency of occurrence among the collected verbs for helpers. A more even balance between speaking and acting verbs is common in both villains and familial characters, though among the former there is a greater prevalence of violently-tinged verbs (such as yelling) and verbs of hate.

These factors–character archetypes as determined through the prevalence of certain adjectives and verbs–constitute what I call “characterizations” in this project. Now as will be discussed in my clustering section, in my collections I noted an issue that was consistently arising in some of the data after these aspects were in place: a subset of the dataset (approximately half of it) had an overcount of characters due to multiple references to the same character appearing. This presented a considerable problem in some instances, greatly changing the apparent landscape of a tale–for example, for one story suggesting that there are in fact three heroines. To address this issue, k means clustering was utilized to cluster the collected characterizations to group together references to the same character with the expectation that their adjectives and verbs are similar.

With this additional optimization technique implemented, we yielded a more accurate characterization for each of the main characters. Once these characterizations are in place for a given text, we can see which archetypes are present and which are absent and thus the basis for categorizing that story. The model summarizes a text then insofar as it defines the characteristics in terms of which it may be grouped with other tales with similar characteristics, thus telling the reader or researcher the type of story which is at hand. In this way, the project lays important foundational work for further research into automatizing the identification of characters as certain archetypes and thus further categorization of stories.

Background

As I have previously mentioned, the research into using natural language processing techniques with fiction is limited. But there is certainly growing interest in this genre, particularly in respect to fairy tales, as El Maarouf et al. 2012 points out. Before looking at the specifics of natural language processing work on fairy tales, I will give a brief sense of the general NLP work on summarization. I will then track through key areas of study I have noted among work on fairy tales generally before turning to the very minimally studied summarization of fairy tales. Finally I will discuss a few categorization efforts pertaining to characters in general.

From El-Kassas et al. 2021 and Nazari & Mahdavi 2019, we obtain an overview of the current state of automatic text summarization. They catalog the advances made in automatic text summarization and the different approaches that have been taken in approaching this topic. They describe the two main categories of extractive and abstractive summaries, the former extracting sentences with high probabilities of being important for capturing key facts and the latter forming summaries through generating text based on the text at hand. Nazari & Mahdavi 2019 focus also on comparing single text summarization methods and multi-text. Both point to the importance of this research lying in the excess of information on the internet today due to the constant creation of new texts posted online. El-Kassas et al. 2021 points to the current focus on extractive methods and the necessity to turn attention to abstractive ones, as well as hybrid methods.

With the state of affairs in text summarization in mind, let us turn our attention to the general topic of NLP work on fairy tales (and fiction more broadly). Among the research that has been done into using NLP techniques on fairy tales, we can identify, in addition to mere summarization, three key areas of study: the utility of fairy tales for emotional analysis; their utility for educational purposes; and the creation of infrastructure for more work in the genre, specifically on corpus creation and annotation. Alm & Sproat 2005 and Volkova et al. 2010 are notable for their work on emotional analysis with fairy tales. Alm & Sproat 2005 are concerned with the prediction of emotions in text on a sentence level, with fairy tales being of use due to the clear presentation of moral messages to be gleaned from them (in children's versions at least). In this study, they focus on text annotation of a small set of Grimm's stories, and look at trends in a story's emotional development. Volkova et al. 2010 address the problem of assignment to categories of emotion and the creation of more nuanced emotional analysis systems with the aim of showing commonalities in readers' reactions. Their larger purpose is to better understand perception of emotion in fairy tales in order to advance the emotional analysis field at large. Both utilize Propp's analysis of folk tales in their consideration of the emotion in the story arc.

A dataset of fairy tales is also often utilized in NLP in relation to educational efforts, given that these are a more accessible type of text. We note two examples of such pursuits: the work of Agirrezabal et al. 2018 and the work of Messina et al. 2021. Agirrezabal et al. 2018, utilizing fairy tales due to their common use in education, widely known plots, and easy online access, consider natural language processing techniques for their benefits for extracting linguistic features. Their focus is on generating vocabulary exercises for language learning for students, utilizing primarily WordNet and ImageNet (for the addition of image mapping to content). Messina et al. 2021 is particularly concerned with popularizing NLP among students through a focus on NLP-based games using fairy tales as their text basis.

Finally, a key area of study focuses primarily on corpus creation and annotation for fairy tales. This infrastructure is crucial to the possibility of making further strides in using natural language processing techniques for fairy tales. Even in other categories of study (particularly Alm & Sproat 2005) this appears as a central first step before further work may be done. Four notable projects on the topic are El Maarouf et al. 2012, Lobo & Matos 2010, Lendvai et al. 2010, and Declerck et al. 2010. El Maarouf et al. 2012 points to the rise in interest in NLP research on fairy tales but notes the present lack of resources for such work–the small number of linguistic tools tailored for fiction and fairy tales. Thus, their focus is on creating a corpus of such tools for French fairy tales, with annotations of verb dependency relations, resolving pronoun and other name character references, and the semantic roles of verbs. Less concerned with annotations as such, Lobo & Matos 2010 also create a corpus of fairy tales on which latent semantic mapping is used and fairy tales written by the same author are clustered. Lendvai et al. 2010, influenced by Propp, is interested in adding more linguistic information to these semantically-oriented resources.

Declerck et al. 2010 relates more directly to my own work. They are concerned with creating a semi-automatic annotation approach, with a focus on characters and the actions they take part in. They rely on the work of Propp to focus their annotations, in part utilizing his schema of folktale main characters. This focus on Propp and the proposed breakdown of narratives in relation to character is especially in keeping with my own work in this project. Unlike my work, however, they are focused on relating parts of a sentence to the parts of Propp's functions defining archetypal actions in a folk tale rather than actually determining which of

Propp's archetype a given agent fits. In contrast, my emphasis is on the characters, with less attention to the story arc.

More fundamentally, my approach diverges from Declerck et al., and from the other research on corpus and annotation creation that we have been discussing, in not being focused on the organizing and annotating of texts but on a way of summarizing a given text through the definition of certain key features. As I have stated, my emphasis is on categorization, working towards classification of fairy tales into types, rather than seeking to provide short synopses. I will address first some approaches that *do* seek to create these whole summaries for fairytales (Kartika et al. 2020, Droog-Hayes 2019, R. Zhang et al. 2015). But ours is not the only approach that diverges from the aim of providing such synopses. Like my work, others have focused on extracting certain features important to a tale with the potential usage in a future summary, rather than an immediate focus on creation of a whole summary. We may divide this approach into two groups: events (Adolfo & Ong 2019, Lovering 2016, Goyal et al. 2010) and characters (W. Zhang 2017, W. Zhang et al. 2019). I will discuss these in more detail shortly.

On the topic of summaries in the form of short synopses of texts, there is a major focus on bringing coherence and a simulation of human-like comprehension into automatically created summaries. R. Zhang et al. 2015, using fairytales as well as news articles, seeks to create a model simulating the long- term memory of humans. Toward that end, they use a semantic network trained on large corpora to create the communication of information between different parts of the mind and ultimately to produce a short summary comprised of sentences predicted to be important extracted from the text. When evaluated with the ROUGE metric, they obtain higher scores than other current methods. Droog-Hayes 2019, utilizing Russian fairy tales and again drawing on Propp's analysis, focuses on abstractive summarization due to its closer relationship to how a human might summarize a tale. Rather than seeking specific information in a given tale, they are interested in drawing out discourse structure to extract salient points in a story. Diverging from the interest in summaries based on human cognition, Kartika et al. 2020 is concerned with summaries to allow children to digest longer, more complex narratives. They are focused on the genre of Indonesian folklore and create extractive summaries.

Turning to projects focused more on extracting specific elements rather than creating complete summaries, here we find an approach much more in keeping with my own. Lovering 2016 suggests breaking a given story into a series of events, with the future intention of those events being the basis for a model that could learn the importance of an event in a fairytale. Using logical techniques, she creates events as defined by verb predicates with variables of subject and object. While her work is not complete, it suggests an avenue for the difficult task of analyzing events and, ultimately, overall plot in summaries of fiction. Goyal et al. 2010, working with a collection of Aesop's fables, address the automatic generation of plot unit representations of stories with natural language processing techniques and sentiment analysis of those plot units (i.e. events) to determine positive or negative effects on the character(s) involved in the event.

Adolfo & Ong 2019 take another approach to extracting events of a fairytale, also highlighting the complexity of such an endeavor. They address this issue by using extractive summarization techniques to find sentences from which to construct events. They create a story world graph, relating characters via the events they are involved in. Their work is of particular importance to mine in their specific definition of character, extracting main characters as those that most frequently appeared as verb subjects and creating a collection of character names (if there were multiple), gender, adjectives, numbers of mentions, and relationships. My project narrows in on this aspect of their work, with the dismissal of descriptors of gender and relationships and the addition of verbs to get a sense of their actions.

In this same way my work is also more in keeping with the projects W. Zhang 2017 and W. Zhang et al. 2019 which focus on summary through character descriptions. In both, the dataset used is not fairytales but stories from the online story-sharing platform Wattpad. Main characters are determined to be those included in the summary that accompanies stories on Wattpad and both projects identify adjectives as the best way to describe a character. In W. Zhang et al. 2019 both an extractive and an abstractive method are performed and evaluated based on a comparison of the collected adjectives to the adjectives in the summary, with the latter yielding better results. My project presents separate difficulties to that of W. Zhang et al. 2019 given the lack of a pre-existing summary for the fairy tales I use and thus the lack of knowledge of main characters from the outset or a means of immediately determining which adjectives should be collected. For this reason, I have the additional task of how to collect main characters, for which (as I have noted) I used the approach of Adolfo & Ong 2019. I also found adjectives to not necessarily present a full picture of a character and their archetype so I additionally collect verbs to improve upon W. Zhang's work.

Finally, I will briefly discuss a few other projects that are similar to mine in focusing on categorization into archetypes. Fernandez et al. 2015 seek to extract the social network from a fairytale, along with categorizing relationships as positive or negative and among these determining the protagonist and antagonist. The project includes character identification, relationship extraction, sentiment analysis, and further analyses of the collected network. Theirs is a project more concerned with relationships and their sentiment, a path of study that seems better suited to characterization of a story as a given type than to the explicit identification of character archetypes. Still with their aim of identification of protagonist and antagonist from the collected network we can see how it is relevant. Similarly Goh et al. 2014 is interested in these types of identifications, specifically the protagonist. Their approach uses verbs for this categorization, though, rather than relationship-driven information, and thus is more in keeping with that aspect of my own approach. Although their focus is on film rather than literature, Bamman et al. 2013 emphasizes the importance of character in defining a story, as I do, and suggests similar archetypes to those that I propose. A series of lexical classes are defined, which entail the stereotypical actions and attributes of a character of that type. Given that this approach gives the capacity to concretize assignment of a character to an archetype, it suggests a path that might be useful for future extensions of my own project.

It is evident from this survey of the current state of the field that there is still minimal work being done on ways of identifying character archetypes in literature. While my work is certainly in close dialogue with the work of Bamman et al. 2013, the difference in dataset presents clear problems that his model would not address–the scattered usage of names in fairy tales, the particular structure of sentences, the type of archetypes that appear and what best defines them, etc. My project seeks to address that gap, as well as to extend the use of character archetype to the categorization of stories. While I have not implemented an actual method of automatic categorization, I lay important groundwork for fairy tales for the ease of collecting themes and thus probabilities of a descriptor for a given category. In my discussion section, I will discuss in more detail possible ways in which my work can be extended.

Dataset and Text Pre-Processing

Before looking in more detail at the steps in creating the characterizations, a digression is needed to discuss the specific dataset used in this project and text pre-processing steps that were taken. I specifically worked with a dataset of fifty Grimm's fairy tales. While these are tales originally collected by the Brothers Grimm, the brunt of them are obtained from the versions in Andrew Lang's Fairy Books through Project Gutenberg[5], as these are considered the standard collection of fairy tales and are likewise used by Lovering 2016 in her own summarization endeavors with fairy tales. Some of the tales included have also been analyzed by Adolfo & Ong 2019 but due to Adolfo & Ong 2019's use of only five tales I have gone beyond their purview. Furthermore, some tales have been included directly from the Brothers Grimm collection[6] to bolster the dataset to fifty texts. They use a similar language, though in some cases have more violence and are less specifically child-oriented. Both Lovering 2016 and Adolfo & Ong 2019 focus solely on versions of the tales tailored to children in their chosen datasets, due to their straightforward language and greater moral black and whiteness. There appear to be no significant differences, though, in this summarization model's performance on these tales in comparison to those collected by Lang. The texts range from 3566 to 20,348 words (Figure 2 depicts the length of each text by the height of its bar, showing us the trend in lengths over all the data). Most texts fall between 4000 and 16,000 words as we may see in Figure 3 with 11 texts (as we can see from the height of the column) having the most common length range of 6000 to 8000 words. The average length is 10628.46.

[5] Lang 2009, https://www.gutenberg.org/files/30580/30580-h/30580-h.htm
[6] J. Grimm & W. Grimm, https://www.gutenberg.org/files/2591/2591-h/2591-h.htm

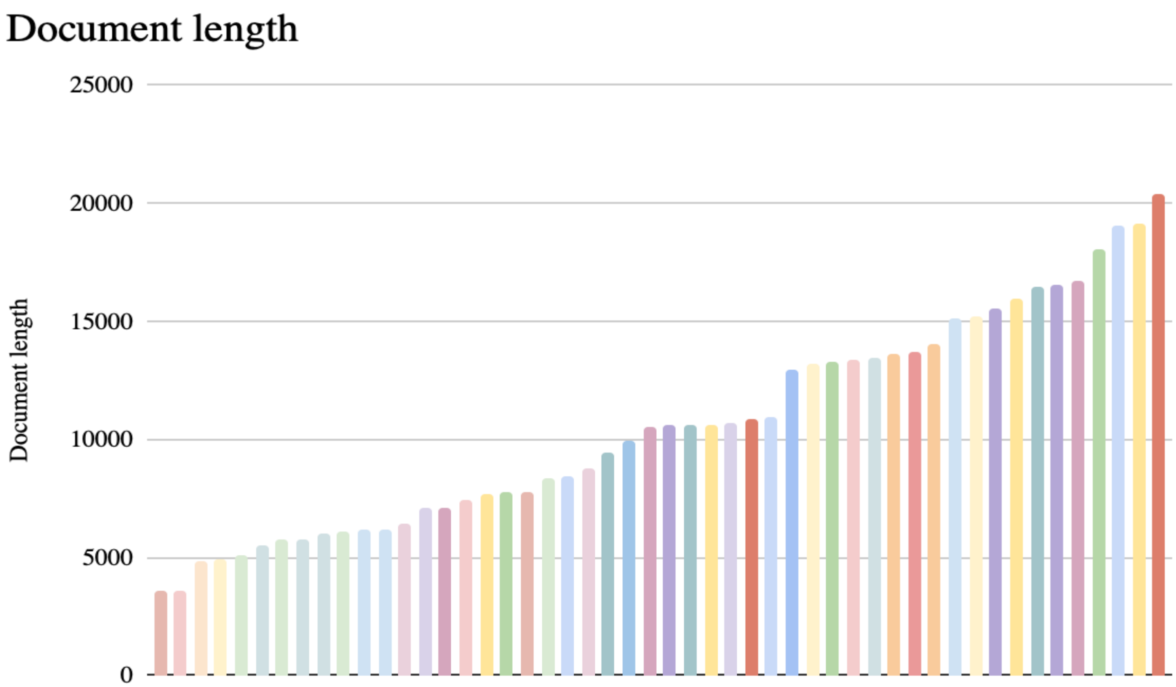

Figure 2: Graph of length of each fairy tale in dataset

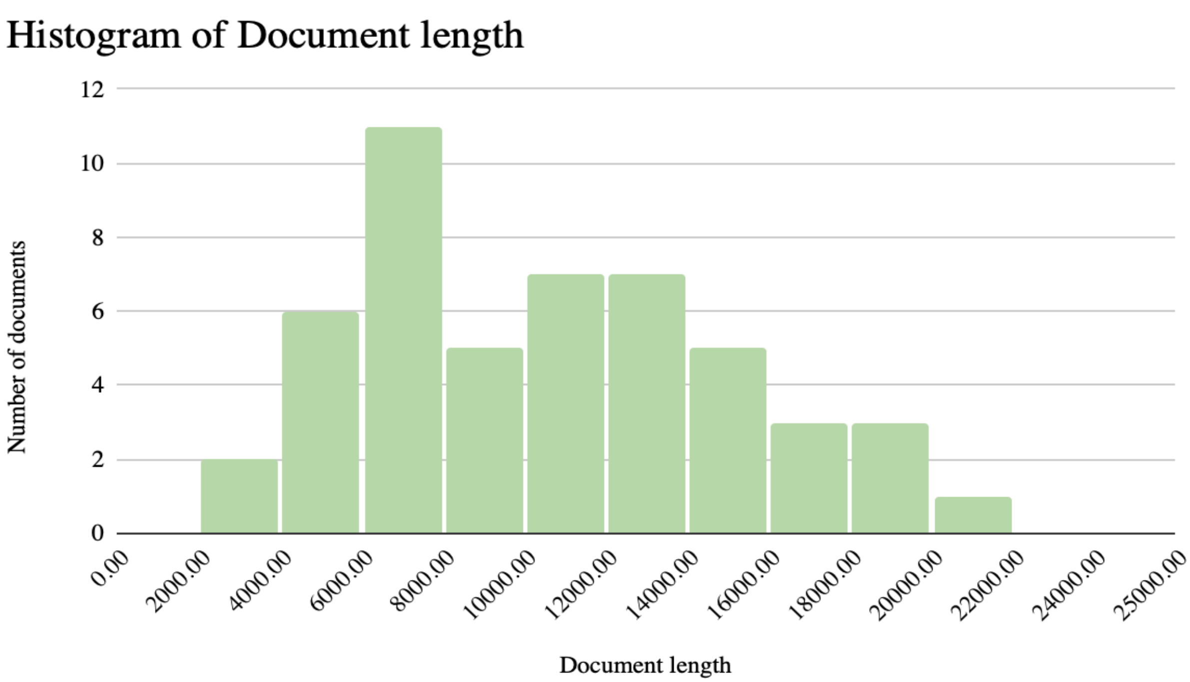

Figure 3: Distribution of document lengths

In processing each text file, it is read in from the path name given in the command line. Blank lines are then stripped out as it is read in. Using the NLTK part-of-speech tagger,[7] a dictionary is collected, keyed with each word and given the values of a list of the parts of speech that word is assigned throughout the document. Likewise with the NLTK tagger, I collect, in the format of sentences, a list of Python dictionaries containing the same information but specific to each sentence in the document. SpaCy's tools are used in addition to NLTK's throughout, which I will discuss at greater length later. The raw text is also put through the SpaCy English language pipeline `"en_core_web_sm"`, which includes a part-of-speech tagger, a dependency parser, a sentence recognizer, a lemmatizer, and named entity recognition, among a few other capabilities[8]. This also prepares the document for the core component of this SpaCy pipeline extension I create which transforms the text.

[7] Bird et al. 2009
[8] Honnibal & Montani 2017, https://spacy.io/models/en

Methods and Results

With this consideration of the processing, we may turn to how specifically I approached creating characterizations. The necessary tasks, as I have laid out thus far, include collection of characters, extraction of adjectives modifying them, and collection of verbs to which they act as subject or object. These tasks allow for a characterization of all main characters in a text to be retrieved which may be used for classifying each character as a Proppian archetype. Secondarily there is the task of clustering in order to group together multiple references to the same character.

First of all for the characterizations, there must be a way to pick out the main characters of a text (and with this task the consideration of a definition of “main character”). Secondly, their qualities (via adjectives) must be extracted, both those directly modifying the character’s name and ideally those at a greater distance in descriptor sentences, particularly those with copular verbs. Finally we must collect each character’s associated verbs, recording both those they are subject of and those they are object of (in that case in addition to the subject of that verb). With these tasks in place, clustering may be performed to group all names and descriptors of a given character to best represent the landscape of the story in terms of the characters and what I will suggest of their archetypes based on the descriptors.

Character Retrieval

To begin to work towards a group of character descriptions, a list of said characters needs to be collected. Before deciding how to accomplish this, though, there needed to be a quantifiable definition of what it is to be a main character. I define a “character” as an entity who is capable of performing actions that affect other characters, the world that the story is set in, or subsequent actions or events. I further limit this to the main characters of a story, which, given the definitional importance to the term "character" of performing actions, is defined as those characters who appear most frequently as subjects of the verbs in a text. Their repeated appearance as a verb subject suggests the performance of multiple actions and thus a continued effect on the trajectory of the tale.

Once a definition was in place, there was another clear problem that had to be addressed: the lack of names for most fairy tale characters. In fairy tales, characters often do not have names but are simply referred to as "the man" or "the woman." A female main character may be named (such as Snow White or Rapunzel) but take the example of *Cinderella*: besides the heroine[9], generally no other character is given a name per se. One might expect a name of the prince but even this is not the case in many of the original versions. Rather, besides Cinderella, the main characters are the sisters, the godmother, and the prince (or Prince at times).[10] This immediately presents a significant problem of how then to collect the most frequently appearing characters, as how are we to differentiate these rather random nouns referencing characters from those referring to another object in the story world? This fact meant that I could not simply use tools that implement Named Entity Recognition, as it is dependent on seeking proper nouns, specifically those that are capitalized for its 'Person' identifier. I followed Adolfo & Ong 2019 in my approach to addressing this issue by taking as main characters the most frequent subjects of verbs. Beyond the supposition that this would lead to returning primarily central characters to the

[9] It is worth noting that, as we might glean from the English translation, the name, Aschenputtel in the original German, essentially means "ash-girl." Still in both the English and German versions, it is treated as a proper noun as we may see through its capitalization.
[10] I take this example from my dataset of tales in which the step-mother is a background character much more so than in our modern versions of the fairy tale.

narrative, I also supposed that, if a noun appears more than once as the subject of a verb, it is unlikely to refer to an inanimate object and thus would likely name some kind of living entity.

I found that selecting as characters nouns that appeared twice generally captured all characters but was far overpopulated, selecting far more random, inanimate object nouns, whereas setting it to four, while usually capturing the hero and villain, generally missed the important side characters such as helpers or non-villainous family members. For example, the main characters of *The Frog Prince* are the Princess and the frog which are exclusively captured when the name occurrence is set to three. When the occurrence was set to two, though, it additionally included the ball and the coach, certainly not main characters in the tale. In *The Peasant's Wise Daughter*, by contrast, when the occurrence is set to four or more the daughter herself is not recorded as a main character. By recording as characters those nouns that appeared thrice or more, there appears to be the best balance between overrepresentation and omission of characters.

To extract the subjects of verbs, SpaCy's dependency parser was utilized.[11] A dependency parser lays out the grammar of a sentence, defining the relationship between "headwords" (generally verbs) and their dependents. We can obtain dependencies such as words that directly depend on the verb (subject and object are examples) and connecting words such as 'and' that nouns depend on. We can also obtain each word's part of speech.

To visualize the information gained, take the sentence from *Hansel and Grettel*:

Hansel and Grettel sat down and ate their bread.

[11] Honnibal & Montani 2017, https://spacy.io/api/dependencyparser

We can view SpaCy's dependency parser representation as a graph, with words as the nodes and dependency relationships as the edges. From this sentence, relationships such as Hansel being the subject of 'sat' and being connected to Grettel via the conjunction 'and' are obtained (Figure 4). The phrase 'their bread' is the direct object of 'ate' (Figure 4). Thus we can obtain the particularities of the grammar of any given sentence with the dependency parser.

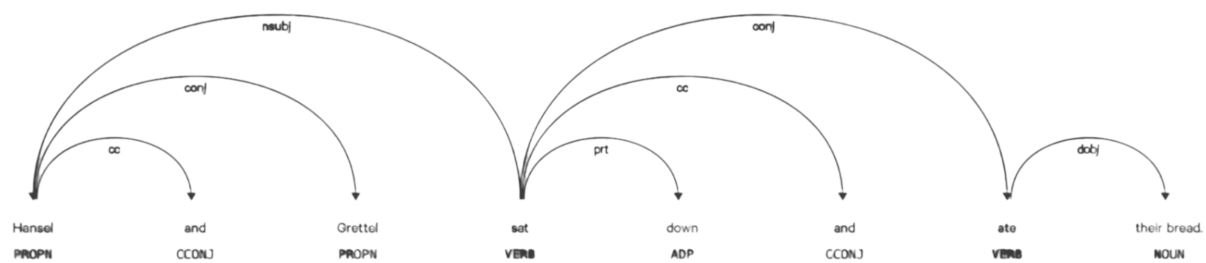

Figure 4: Visualization of SpaCy's dependency parser

To find the desired characters, SpaCy's dependency parser's "chunking" feature was used, which breaks the sentence into noun phrases, such as simply a single noun or a noun and its adjective modifier(s), and records the root–generally a verb or preposition that relates an indirect object to a verb, such as 'toward'–of each phrase (or chunk). Different chunks can have the same root.

```
for chunk in doc.noun_chunks:
    if chunk.root.dep_ == 'nsubj' or (chunk.root.text[0].isupper()):
        if stopWordTypes(chunk.root.text,dictionary):
            continue
        else:
            if ''' not in chunk.root.text and ''' not in chunk.root.text and
len(chunk.root.text) > 2:
                chars[chunk.root.text] = chars.get(chunk.root.text,0) + 1
```

Iterating through the noun chunks of a text, each chunk is checked for its dependency upon the root. If it is a subject of the verb, or secondarily if it has a different dependency but is capitalized, the word continues to be considered. The latter is included so as to not miss more easily identifiable character names and their mentions even if they are not verb subjects. Then it is determined whether or not it is a stop word. The stop word method returns a boolean value of whether or not the given word is a stop word, defined as types of pronoun and types of articles.[12] If it is not a stop word, the word is checked for any stray apostrophes in it, as I found a persistent trend among my dataset of misidentified words because of misplaced punctuation and thus consistent typos coming through as characters. If the chunk does not include one, the word is finally recorded in a dictionary as the key and one is added to its value to update the number of times a certain word has been seen. Once all noun chunks have been checked, the collected dictionary is iterated through to record only those entities that have a value of three or more (given my previously stated reasoning), thus having occurred three or more times in the text as verb subjects or capitalized words.

Results of Character Retrieval

With this model, the main characters of each text could be obtained. To determine the accuracy of such retrieval, though, it first needed to be determined how many characters were in each text and who they were. I collected these sets for each document, determining which

---

[12] No coreference resolution was used to link such pronouns to the noun to which they refer as, though I briefly utilized the NeuralCoref module, the standard coreference resolution model in Python, it performed quite poorly on the fairy tales likely due to their at times odd sentence structure. Others that implemented coreference resolution for fairy tales or fiction more generally utilized the Stanford CoreNLP model (or a subset of it, BookNLP, specifically for longer documents) to good effect (Adolfo & Ong 2019, Lovering 2016, W. Zhang et al. 2019). While this ended up being outside the scope of this project, a future step to improving character collection, as well as adjective and verb extraction even more so, would be to implement such a coreference resolution.

characters should be retrieved. This exposed an issue that arose in the model's character collection as I mentioned previously: each character, whether named or not, is generally referred to by at least two or three different names. Say there is a character referred to as 'the woman.' Then she likely will be referred to also as 'the wife' or 'the mother,' perhaps in equal measure. Thus while the model might collect ten seemingly separate entities, there may in fact be only five unique characters in the set, simply referred to by different names. In my hand-collecting of characters, then, I tracked all of the names used to refer to a given character to best determine the accuracy of the character retrieval. I found there to be on average 5.1 characters per story, with a standard deviation of 1.62. The most common number of characters was 6 with nearly 15 texts and most texts fell in the range of 4 to 6 characters, as we can see by the three tallest columns (Figure 5).

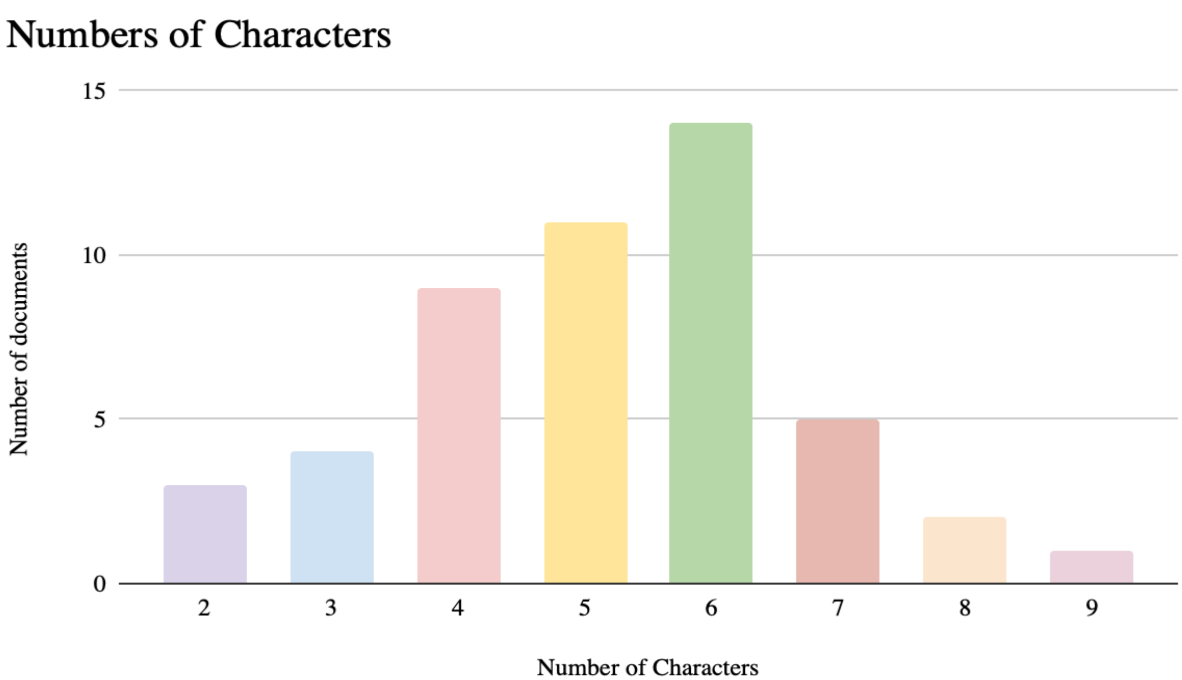

Figure 5: Distribution of numbers of characters among the documents (N=50)

With the knowledge of the characters that are present in each story, the accuracy of the automatic retrieval may be determined. The performance was determined by precision (correctness among the collected set), recall (how many of the characters that should be included are included), and F1 (the balance between precision and recall) scores for each document.[13] On average, there was a precision of 0.88, a recall of 0.91, and a F1 of 0.89. We may see the different numbers of documents with the certain scores across all documents in Figure 6, gaining a better understanding of the most common scores for each metric. It may be noted the recall has the fewest dips in performance and never falls as low as precision. Focusing in on F1-scores, since it balances precision and recall, we may see that most scores fall between 0.80 and 1.00 by a large margin, given the three, by far tallest columns occur in this range (Figure 7). In fact, only five texts have a score (precision, recall, or F1) that falls below 0.60, with only one text having an F1-score below 0.60. No obvious themes arise between these texts in terms of variation in length and number of characters.

[13] To calculate these scores, I am including as true positives all names/references that refer to a main character as determined by the nouns I likewise included in the set of names for a given character in hand-annotating the documents.

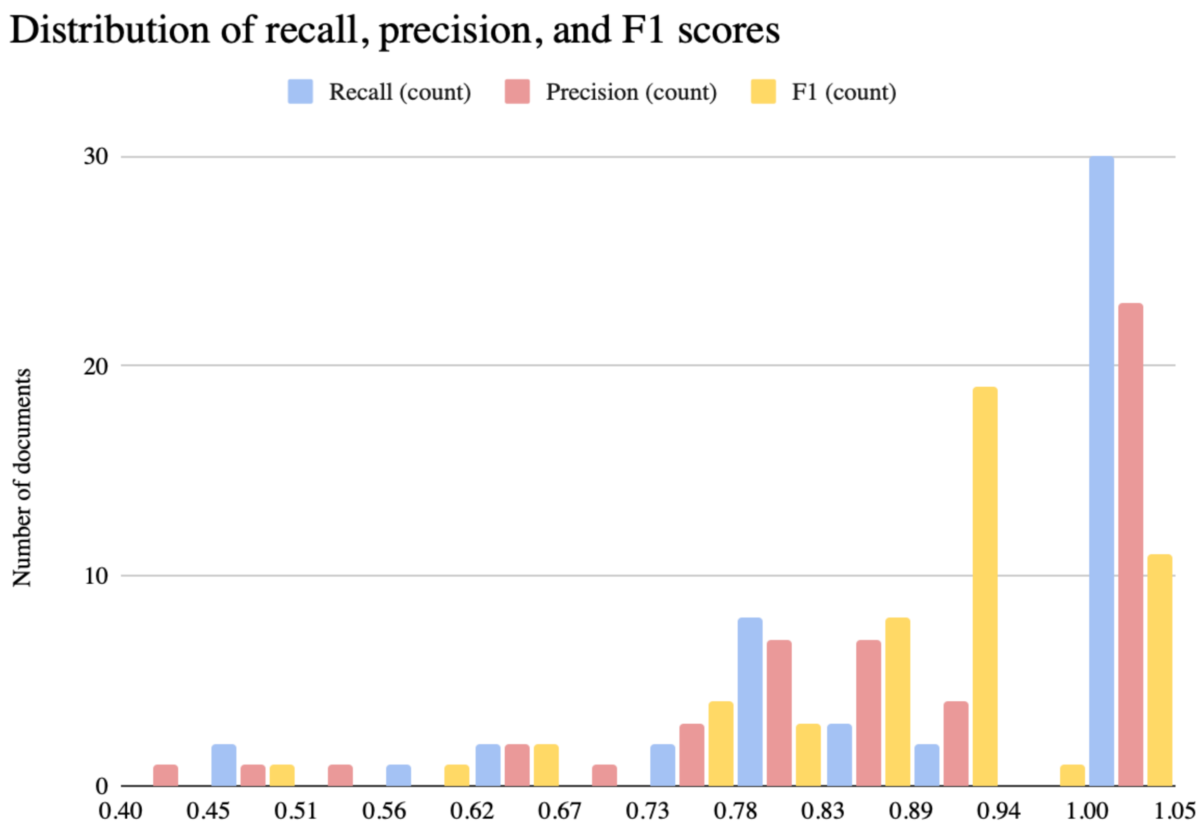

Figure 6: Distribution of document's precision, recall, and F1 scores across the dataset

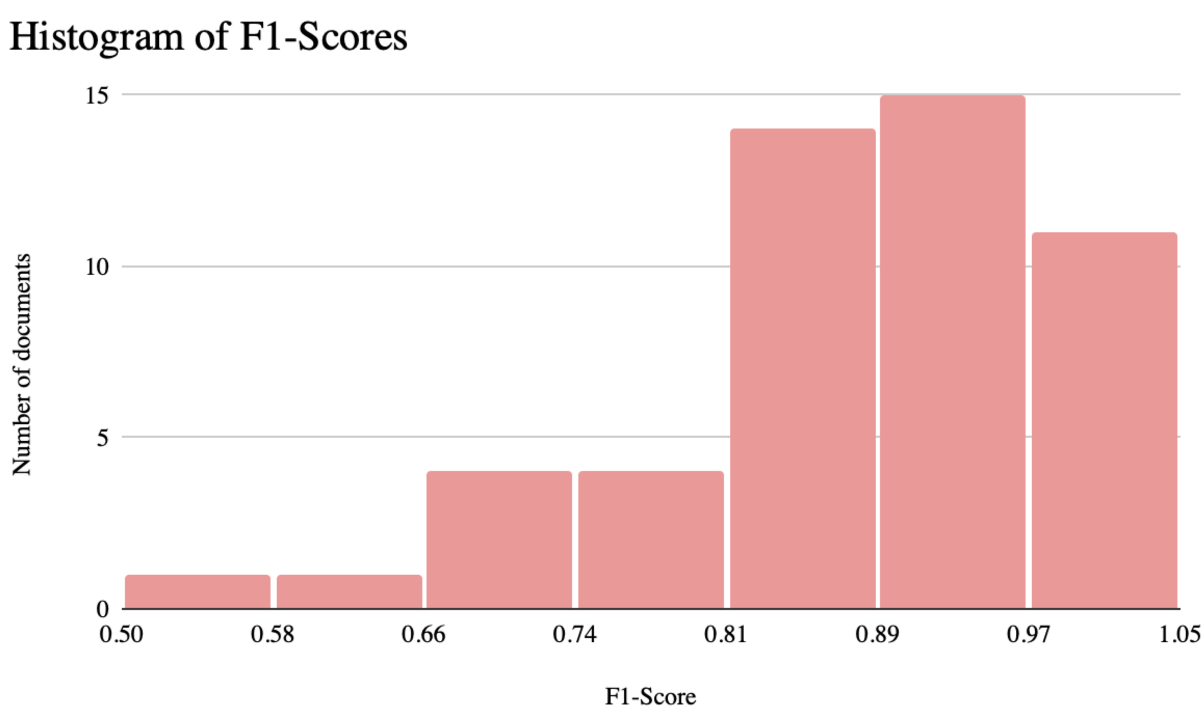

Figure 7: Distribution of F1-Scores for document character retrieval

Adjective Extraction

With a collection of characters in place, adjectives for each character must be collected to form the actual characterizations. For this task, it must be considered when most descriptors of a given character appear. One clear place is an adjective appearing in a noun phrase with the given character noun, where it is clearly modifying that word. A second type of adjective describing a noun is a predicate adjective, where the noun and its adjectives are separated by at least one word, namely the verb. While this can occur with any verb, it is most common with copular verbs (forms of 'to be' for example), which W. Zhang et al. 2019 likewise notes as he targets sentences with copular verbs for character attribute extractions. The second case is a bit trickier, though, even given the limitation to copular verbs, given the lack of knowledge of where an adjective describing a character may appear in the sentence in relation to distance from the character. When giving examples of adjective modifiers, often the first case I describe is used (with an adjective right before its noun), where through the spatial relationship we can easily infer in English that the adjective likely modifies the verb (for example, in the phrase "the beautiful princess" the adjective is right before the noun). There also may be an example falling in the second case for adjectives with a copular verb. Often, though, it will be a simple sentence that also shows some clear spatial relationship between the adjective and noun, namely with only the verb separating character and adjective or adjective phrase (W. Zhang et al. 2019 gives the example of "Bill is not evil," with "not evil" being the sought adjective phrase).

However, a sentence structure this simplistic is not often the case in practice and particularly not in the genre of fiction or fairy tales. There is often a greater linearity in the organization of words in a news report or academic paper, data on which NLP models are more often trained. While fairy tales may have a simplicity in the types of words that are used and in general with their sentence structure, the order of words is often more complex. For example, take the first sentence of *Hansel and Grettel*:

> Once upon a time there dwelt on the outskirts of a large forest a poor woodcutter with his wife and two children; the boy was called Hansel and the girl Grettel.

Here we have the classic opening 'once upon a time' but then we have the direct object, then the verb, then a subclause describing the direct object, and then finally the sentence's subject, 'a poor woodcutter.' Here we see how the sentence is simple in the words used and what it describes but the syntax makes it more complex. We may see this with adjectives too. This sentence from *Cinderella* describes the heroine thus:

> However, Cinderella, notwithstanding her mean apparel, was a hundred times handsomer than her sisters, though they were always dressed very richly.

We learn here that Cinderella is handsome but again there is an intermediary phrase with another description, as well a qualifier ('a hundred times') on the comparative adjective.

This is all to underline the potential complexities of extracting adjectives in the given dataset. For this reason it seems, I found very poor results from extracting adjectives with SpaCy's dependency parser and its 'AMOD' relationship. Its accuracy in the `"en_core_web_sm"` should perform somewhat less accurately than the two larger pipelines, the most high-powered having a 95.1% accuracy and the other a 91.0%, but near to them.[14] On this data set, though, it had less than 10% accuracy. In contrast, W. Zhang et al. 2019, working on extracting characterizations from fiction, particularly stories from the online platform Wattpad, uses the more state-of-the-art Stanford CoreNLP library, specifically its adjacent BookNLP model, to

---

[14] Honnibal & Montani 2017, https://spacy.io/usage/facts-figures

extract adjective modifiers. Using two models, one extractive and one abstractive, he uses tools from BookNLP to extract adjectives and still only obtains approximately a 25% recall accuracy with the former and a 50% accuracy with the latter.

Because of these difficulties, I did not utilize a specific tool for determining adjectives modifying a given character's noun reference. Rather I decided to define an adjective as a modifier as dependent on its distance from the given noun. First, we must define a distance. Specifically considering adjectives appearing in sentences with the character's name or associated noun, it is set that adjectives between two words prior to the character name and up to eleven words after will be recorded. Obviously this may lead to an overcollection of adjectives but it is with the assumption that *most* adjectives that appear within a sentence with a copular verb and a given noun as its subject will be in reference to that noun. Eleven is specifically chosen as I found that on average, based on a set of example sentences, adjectives in that range had the highest probability of referring to the given character reference compared to larger numbers. This does in practice of course lead to overcollection and thus creates a lack of precision outweighing the potential for better recall. While many adjectives do occur at a shorter distance, a large enough percentage on average appears at this greater distance of 11 that the aid in precision a smaller distance would yield is outweighed by recall issues. Anything smaller greatly decreased the number of adjectives gathered, missing a great deal of key descriptors. Furthermore, the two words prior to the character reference are checked for being adjectives to capture the direct modifiers in a copular sentence.

The code is as follows (in the context of the overall iteration through a list of sentences determined to have a given character name in them with each word in a tuple with its part of speech):

```
if ('was', 'VBD') in t or ('were', 'VBD') in t or ('are','VBZ') in t or
('are','VBP'): #checks if verb of sentence is form of 'is'
                    if pos == 'JJ' or pos == 'JJR' or pos == 'JJS': #adjective
check
                          if char_posit[1] == True or char_posit[0] > 0:
                              if count >= (char_posit[0])-2 and count <=
(char_posit[0] + 11):
                                  if not word == '"'
                                     and not word == "'"
                                     and not word == '"'
                                     and not word == "'"
                                     and ('.' not in word)
                                     and ('.' not in word):
                                     if len(word) >= 2:
                                          adjs.append(word)
```

Sentences that specifically have past and present third-person verbs are targeted as we only wish to consider sentences where the character name could be the subject. Besides determining if the sentence has a copular verb, the current word token's part of speech is examined to see if it is an adjective and if it is occurring in the set boundaries around the character name. As well the word is examined to see if it neither is punctuation nor has punctuation in it, and if it is not shorter than two characters. In addition to this collection of adjectives, all sentences in which the given character is present are checked for adjectives coming before the name, specifically determining if the current word is the character name and if the last word (stored at the end of each iteration) was an adjective.

```
if (last_wrd[1] == 'JJ' or last_wrd[1] == 'JJR' or last_wrd[1] == 'JJS' or
last_wrd[1] == 'CD') and (word == char): #checks if, if not 'is' verb, if there is
modifying adj
   adjs.append(last_wrd[0])
```

The resulting list of adjectives is finally sorted alphabetically and then returned.

Verb Extraction

Before discussing the results of the adjective retrieval, I will discuss my method of verb retrieval so that the results of both may be put in dialogue. I diverge from W. Zhang et al. 2019's method of characterization which relies solely on adjectives in also including verbs for each character to complete their characterization. While there are rather clear cut descriptors of characters throughout most fairy tales, I hypothesized that more depth about each character could be gleaned by a consideration of their actions and the actions done unto them throughout the story. Through verbs, we may get a sense not simply of the looks or moral character bestowed upon an individual by the author or by other characters but how they affect the course of events and how they are affected. This approach allows us to better characterize not simply the individual character but also the type of story that a given story is, as it defines what sort of action occurs and what sort of characters populate it. This potentially differentiates it from other versions of the same folktale where the same character may act differently. The adjectives and verbs may color each other, as well: certain kinds of actions, such as 'speaking' and 'advising,' might suggest an adjective like 'old' refers to a mentor rather than a villain; similarly, the adjective 'wicked' could suggest a character is in fact a villain despite the verb 'crying,' which might otherwise suggest an aggrieved hero. The two create a more fully formed representation of each character and thus of the story world they make up.

To go about this verb extraction, SpaCy's dependency parser was once again utilized. The tokens of the document are iterated through, tracking each new verb to designate a new phrase and recording it in a dictionary (keyed with verbs with values of a list of their subject, object(s), and indirect object(s)). If it is found to include a given character as its subject, object, or indirect object by the time a new verb is found, it is retained; otherwise it is removed from the character's dictionary. The core of it is as follows:

```
        if token.head.pos_ == 'VERB': #check if head word is verb
            if not token.head.text == verb: #check if it's a dif head word than
current, thus a new phrase/sentence
                if not charD.get(verb, []): #if nothing was added as val, continue
                    continue
                else:
                    if char not in charD.get(verb, []): #if char never nsubj, dobj,
or pobj of verb, delete it from dic
                        del charD[verb]
                verb = token.head.text #update verb to new phrase's
                if verb[0].isupper():
                    verb = verb.lower()

        if token.dep_ == 'nsubj' or token.dep_ == 'dobj' or token.dep_ == 'pobj':
            charD[verb] = charD.get(verb,[]) + [token.text, token.dep_] #add tok and
its dep
```

It is determined whether the current token has a direct relationship to a verb and, if so, whether it is a new verb, designating a new phrase. This determines the end of the last phrase has been reached, so the list value of the verb in the dictionary is checked to see if anything has been added and, if it has, the list is checked for the given character. If they are not found, the verb is deleted from the dictionary. Then the current verb is updated to the newly found one. Finally, the current token is added to the verb's list if it is the subject, object, or indirect object of the verb. For each character, the compiled dictionary is returned and added to a dictionary as the character name's value.

Results of Adjective and Verb Extraction

Retrieving these full characterizations, all characters may be output along with their respective lists of adjectives and verbs. Before considering accuracy, I will lay out basic statistics on what was retrieved. On average, 4.74 adjectives per character were retrieved with a standard deviation of 1.69 and 5.47 verbs with a standard deviation 0.80. In Figure 8, we may see that most texts had approximately 2-6 adjectives and, in Figure 9, approximately 4-6 verbs. There was more variation in numbers of adjectives between texts than numbers of verbs, and both were not greatly correlated in what variation they did have to changes in length or number of characters. For example, the longest text (*12 Dancing Princesses*) had 7 characters, with an average of 5.59 of both adjectives and verbs per character. In contrast the shortest text (*The Giant and the Tailor*) had 2 characters with an average of 7 adjectives and 5.50 verbs per character. While this example suggests a strong correlation between length of text and number of characters, it is important to note that this is not really the case either. There might be somewhat more correlation with characters than with adjectives and verbs but it is still rather scattered when there are greater or fewer characters. We may better see in Figure 10 the lack of trend of verbs and adjectives in relation to number of characters, as there is no trend line of increase or decrease as the number of characters increases.

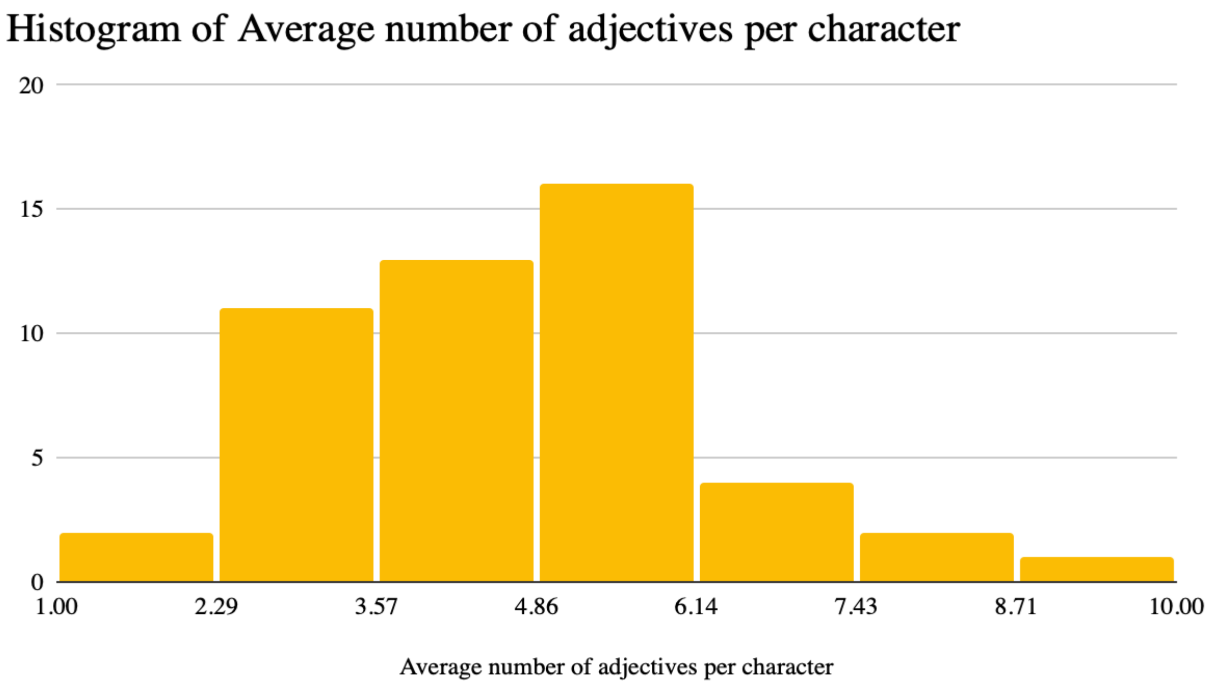

Figure 8: Distribution of retrieved adjectives across texts

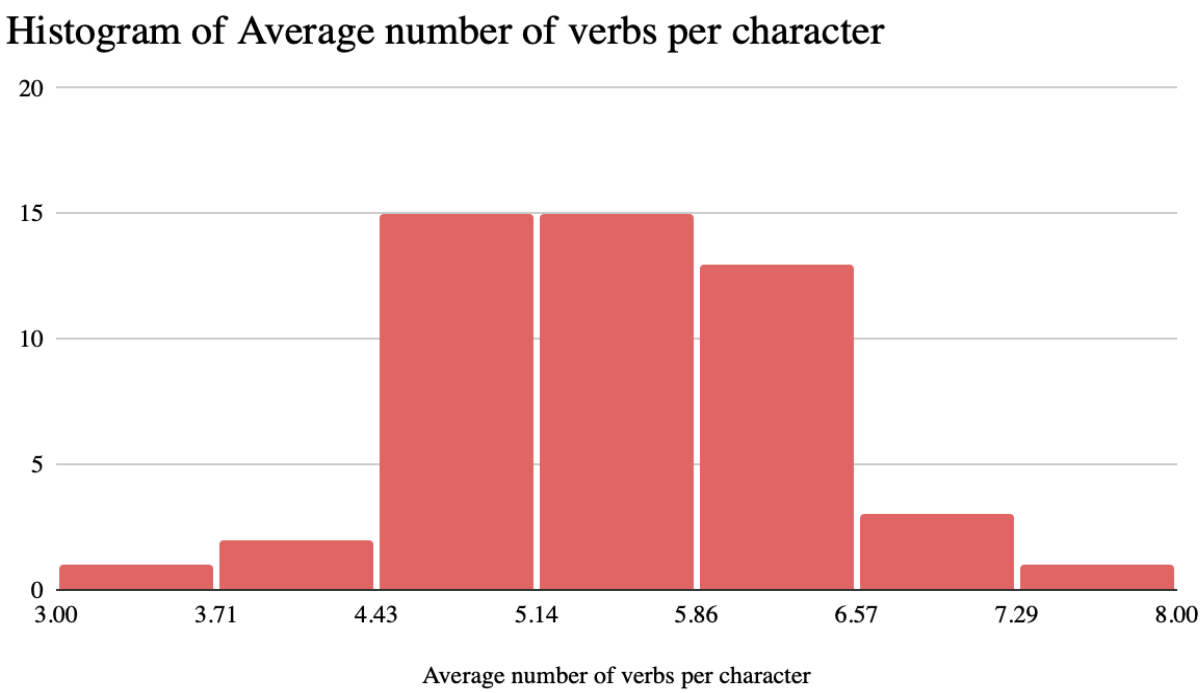

Figure 9: Distribution of retrieved verbs across texts

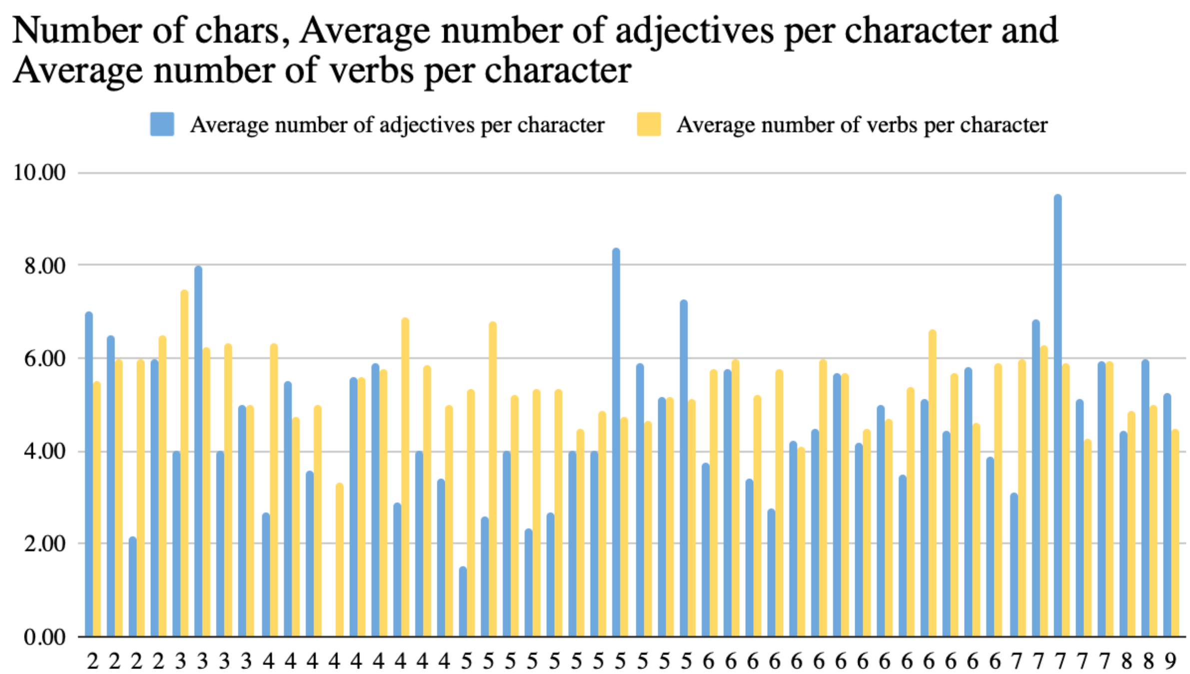

Figure 10: Number of characters and average number of adjectives by number of characters for each document

To calculate the accuracy of these retrieval methods, a subset of the whole dataset was selected, comprised of 5 texts (10% of the whole dataset). The texts were chosen to be best representative of the dataset, ranging from the smaller documents to the larger and with varying numbers of characters. From each text, 10 sentences were selected at mostly random, though fitting the following criteria of (1) representing most or all of the characters in the text; (2) including adjectives directly before some character names and in sentences with copular verbs; and (3) including verbs that characters are subjects or objects of. These sentences were then hand-annotated for the verbs and adjectives associated with an instance of a character's name. For example, we may consider the following sentence from *12 Dancing Princesses*:

That day, when he made up the bouquets, Michael hid the branch with the silver drops in the nosegay intended for the youngest Princess.

The character names, Michael and the Princess, are highlighted, as is ‘hid’ which Michael is the subject of and ‘intended’ which the Princess is the indirect object of. The adjective ‘youngest’ describing the Princess is also highlighted. When this is done for every sentence, we may see for each document what accuracy we obtain, as well as averages to get a sense of the ability of the adjective and verb retrieval at large. On average, adjective retrieval had an F1-score of 0.75 and verb retrieval had an F1-score of 0.66 (more results may be found in Figure 11). The performance of both were particularly good on *12 Dancing Princesses*, the longest documents among the dataset, with an adjective F1-score of 0.80 and a verb F1-score of 0.93. They performed poorly on *The Peasant’s Wise Daughter*, one of the shorter documents, with F1-scores of 0.53 and 0.45, respectively. Here there appears a clearer correlation between length and accuracy; there was overall a trend of more success with longer documents and poor performance with quite short ones, which we may see overall in the performances of *Willow, Wren, and Bear* and *Peasant’s Wise Daughter* in Figure 12.

Document Names	**Adjective Precision**	**Adjective Recall**	**Adjective F1-Score**	**Verb Precision**	**Verb Recall**	**Verb F1-Score**
12 Dancing Princesses	0.81	0.79	0.80	1.00	0.86	0.93
Hansel and Gretal	0.53	0.80	0.64	0.83	0.71	0.77
12 Brothers	0.67	0.83	0.74	0.67	0.56	0.64
Peasant’s Wise Daughter	0.44	0.67	0.53	1.00	0.29	0.45
Willow, Wren, and Bear	1.0	1.0	1.00	0.60	0.42	0.51
Average	**0.69**	**0.82**	**0.75**	**0.82**	**0.56**	**0.66**

Figure 11: Accuracy of verb and adjective retrieval based on subset of 10% of dataset, based on randomly selected set of 10 annotated sentences for each

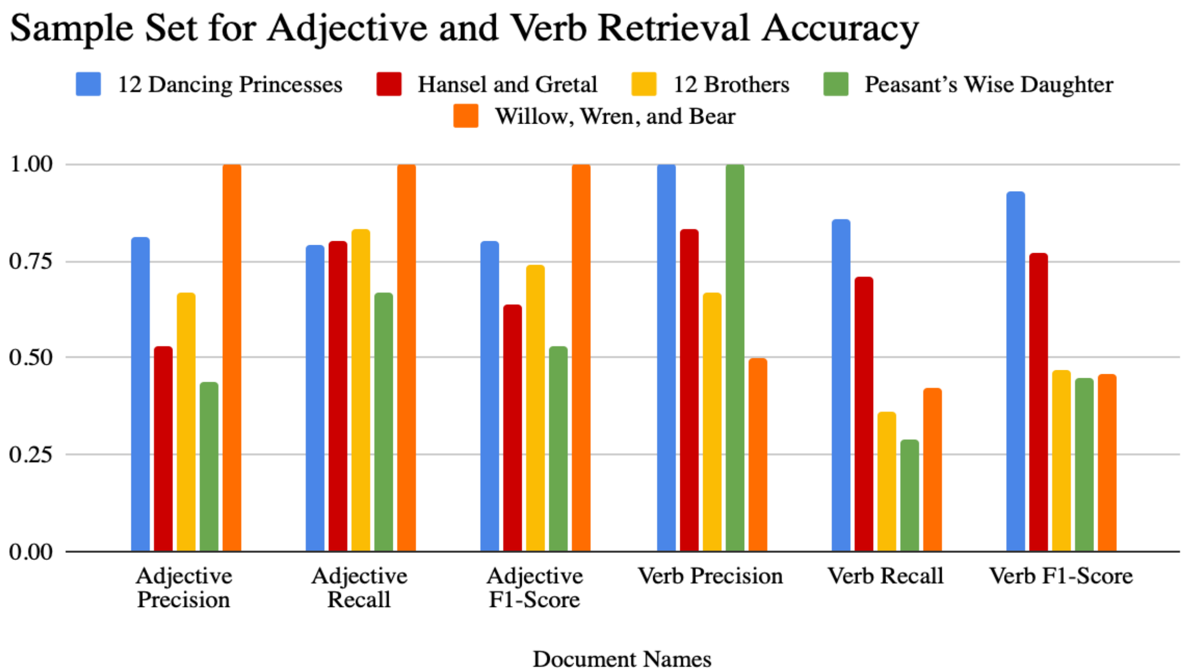

Figure 12: Precision, recall, and F1-scores of adjective retrieval and verb retrieval for each document in sample set

Clustering of Characters

With the adjectives and verbs collected, character descriptions from which we may draw archetypal themes are in place. But the issue I presented earlier lingers–the over-representation of characters in the story due to different references to the same character each represented as their own character in the model's output. With some knowledge of the story (or often simply human inference), one can generally connect which references are in actuality the same character, but the issue remains of automating the collecting of different character references into a single one. Still, we have facts that we might imagine would be shared among different references to the same character–namely, their associated adjectives and verbs. Due to this, we have a list of data that can be compared amongst the set of characters to determine similarity, specifically clusters of similar character information collections. There is the hope that the same characters would have similar characteristics. This situation suggests that creating vector representations of the created description of each character reference among the returned set of entities and then using clustering to group those vector representations might helpfully lead to the different names and descriptions of the same character being clustered into a singular entity. This would serve to both more fully describe that individual character and to give a more accurate count of the set of characters at play in the tale.

Thus, to create a more precise representation of the tale at hand through the collection of character descriptions, I implemented a form of clustering, specifically k means clustering through NLTK.[15] This clustering technique takes in a k value determining how many clusters to create out of the data (specifically vectors here) and then, through a series of iterations, clusters the data around the centroids of the clusters based on each data point's distance from a given centroid as compared to the other centroids. The distance may be calculated as Euclidean or cosine distance but I am specifically using the latter, as the Euclidean distance performed much worse on average, rarely yielding the correct clustering.

Before the clustering can be performed, though, the data needs a few extra processing steps and then a vector representation of each character description must be created. For creating the vectors, I utilized the model `'glove-wiki-gigaword-50'`, a pre-trained version of Glove, a common model for creating vectors, trained on Wikipedia articles with vectors of size 50.[16] While these are rather small vectors, from testing with vectors of sizes 100 and 300 I found minimal improvement in performance from these larger vectors but severe impacts on the speed

[15] Bird et al. 2009
[16] Pennington et al. 2014

of runtime even for the comparatively small documents used in my dataset. Thus the increase in size of vectors did not seem worthwhile in comparison to its costs.

Firstly, then, a list of all adjectives and verbs and the name with which they are associated are compiled into a single list for each character reference within the set of collected characters. Then all words in all lists are made lowercase and all dashes and spaces are removed. This is in order to have the best likelihood of a given word appearing as a key that has a vector representation. For example, the adjective "tender-hearted" appears in a few fairy tales. It is generally written with the dash within the stories but a vector representation of this term does not appear in the Glove model. But the term "tenderhearted" does in fact appear. Capitalization leads to issues with certain character names: some names which are common do indeed have vector representations but only in the lowercase form. I found issues with missing existing vector representations to relate particularly to dashes, spaces, and capitalization so these were particularly important to target in processing.

When these steps have been accomplished, a vector representation of each word may be obtained through the Glove model. But besides a representation of each word, there must be a collective representation for the whole character reference description (the name, adjectives, and the verbs). To accomplish this, the average of each individual word in a description's vector is determined to be the representation of that whole description, as shown below:

```
vecs = []
for sent in list:
   nw = len(sent)
   nv = 0
   for word in sent:
       try:
           word_vec = glvWiki[word]
           nv += word_vec
       except:
           pass
   v = nv / nw
   vecs.append(v)
```

A list of vectors is created and then the list of character descriptor lists is iterated through. For each, its length is recorded. Then each word is tested to see if there is a vector representation of it in the given model. If so, the vector representation is obtained and added to the sum of vectors of words in the characterization. Otherwise, the word is passed by. Once the sum is divided by the number of terms in the characterization to obtain the average vector, it is recorded in the list of character description vectors.

Once an average vector representation has been calculated for each character reference and their descriptors, these vectors may be clustered. Given a k value, the vectors may be clustered into k groups, a process defined like so using the NLTK k means clusterer:

```
input = int(sys.argv[2])
kClust = input
kclusterer =
KMeansClusterer(kClust,distance=nltk.cluster.util.cosine_distance,repeats=25)
assigned_clusters = kclusterer.cluster(vecs, assign_clusters=True)
```

The k value is obtained from command line input as it will be specific to each text. Then for the clustering the model is set up with the type of distance to be used (cosine in this case) as well as the number of repeats at each iteration. Then the actual clustering is done with the input of the average vectors for each characterization.

There is an important question for the clustering that I have not yet discussed though: how to determine the value of k (the number of clusters) to best cluster each set of data for each text. Ideally, the value of k should match the number of characters known to be in the story. But it also must take into account (1) not exceeding the number of character references actually retrieved and (2) leeway for potential extra nouns wrongly included among the character references. I will discuss the results of clustering the benefits of different assumptions around k values. As a baseline, I initially took the number of characters we know to be in the story as k and, if there were not enough character references to match that, decreased k.

Results of Clustering

To test the results of the clustering, I took a subset of one third of the data with a character overcount, yielding a dataset of 9 to test the results on. The dataset is representative insofar as covering the breadth of document lengths and numbers of characters in the original set, as well as variation in numbers of characters with extra references and numbers of extra references. The baseline of accuracy was determined by calculating precision, recall, and F1 scores for setting k to the value of the true number of characters in the story or, if that was not possible, the closest possible value. Given this approach, on average the F1 score was 0.72. While this performance was not utterly inaccurate, this assumption of k value as might be expected due to the factors I previously discussed does not yield ideal results.

With this approach in mind, I compared it with a different approach of determining k using best fit, the number of lines to best represent a curve, on a graph of k value vs distance from cluster centroid. Specifically, for each document, the distance from the closest vector in each cluster to that cluster's centroid was determined using cosine distance for each value of k between 1 and the total number of characters retrieved.

```
clustMeans = kclusterer.means() #cluster centroids
closestVecs = {}
for index, vec in enumerate(vecs):
   aC = assigned_clusters[index]
   curDist = nltk.cluster.util.cosine_distance(vec, clustMeans[aC])
   if aC in closestVecs:
       if curDist < closestVecs.get(aC, 0):
           closestVecs[aC] = curDist
   else:
       closestVecs[aC] = curDist
```

Once the distances were obtained using the above means, the average of each group of distances was taken to be the distance for that value of k. Then these distances and their associated k values were plotted to find the curve (an example with *Blockhead Hans* may be seen in Figure 13). I determined the value of best fit by eye. Thus for each document there was a new value of k to be utilized.

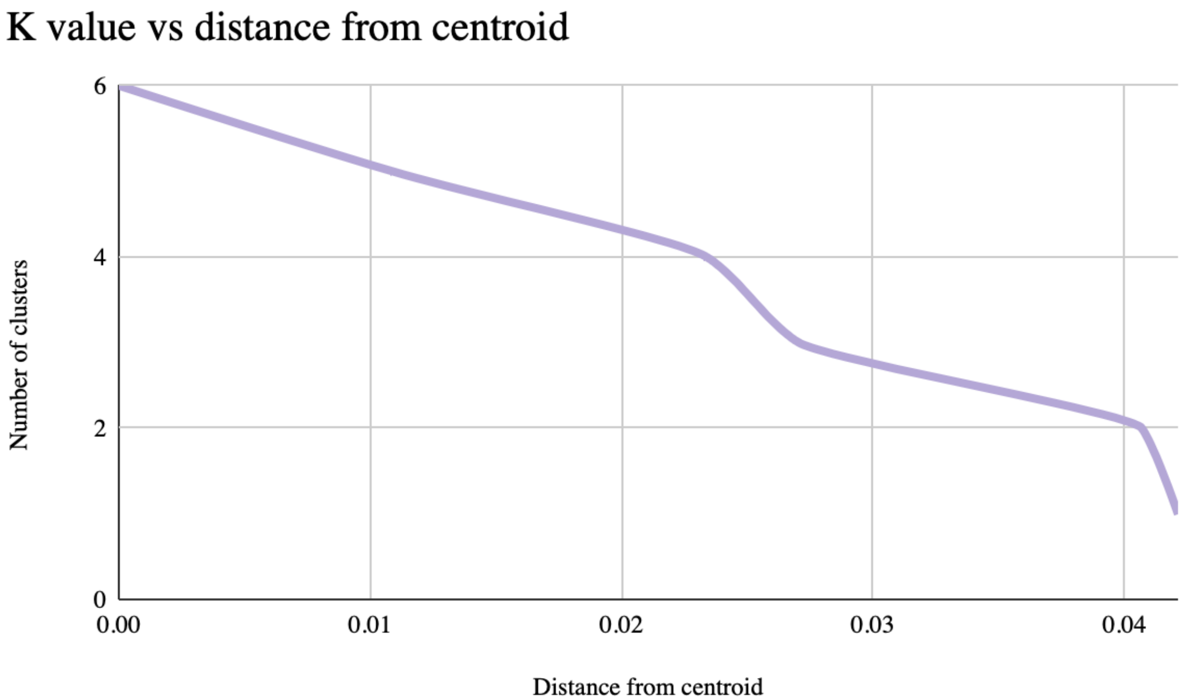

Figure 13: Plot for *Blockhead Hans* of k value vs average distance from a cluster's centroid

This second approach did indeed yield better results with an average F1 score among the dataset of 0.89, thus a 17% increase in performance. With k = number of characters, about half of F1 scores fall below 0.7, with the worst score being 0.45 on the tale *Blockhead Hans*. With k = best fit, on the other hand, no score falls below 0.7 and only 2 fall below 0.8, less than a third of the data. We may see the overall scope of performance for F1 scores across the dataset in Figure 14 and the performance of precision and recall in each text in Figure 15 (here blue lines correspond to precision and yellow to recall so that their individual increases with best fit instead of character number may be more easily seen).

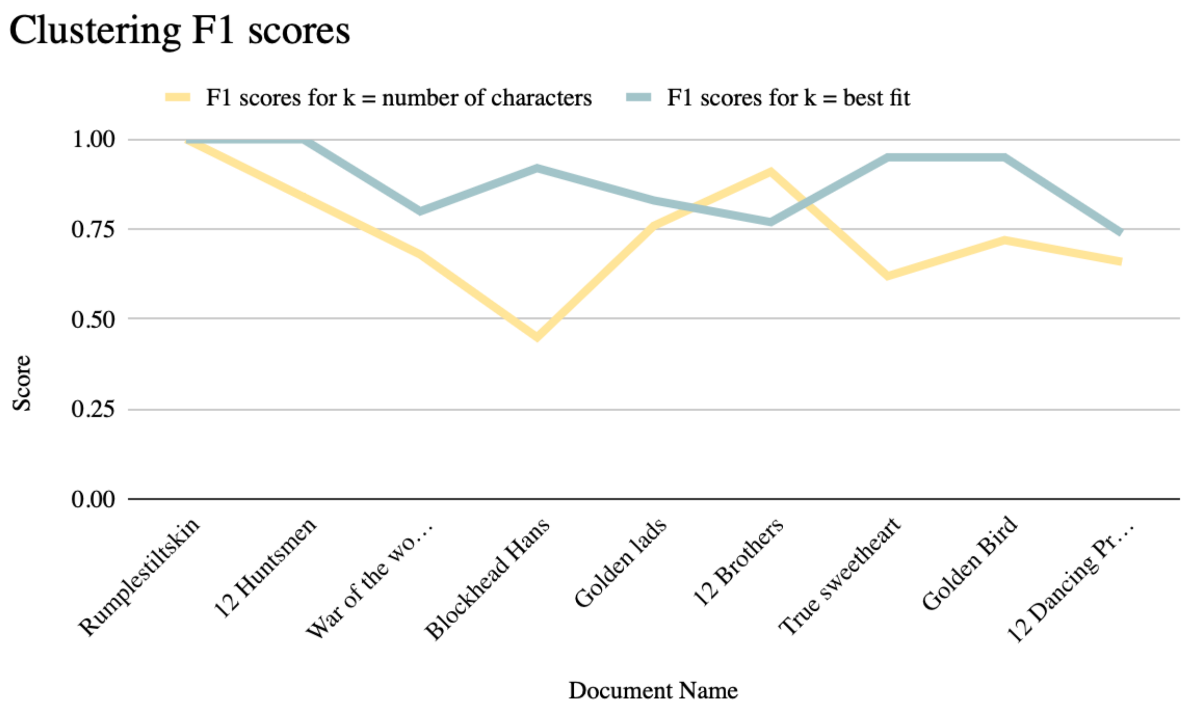

Figure 14: F1 scores for clustering of each document (N=9) with k = number of characters vs k = best fit

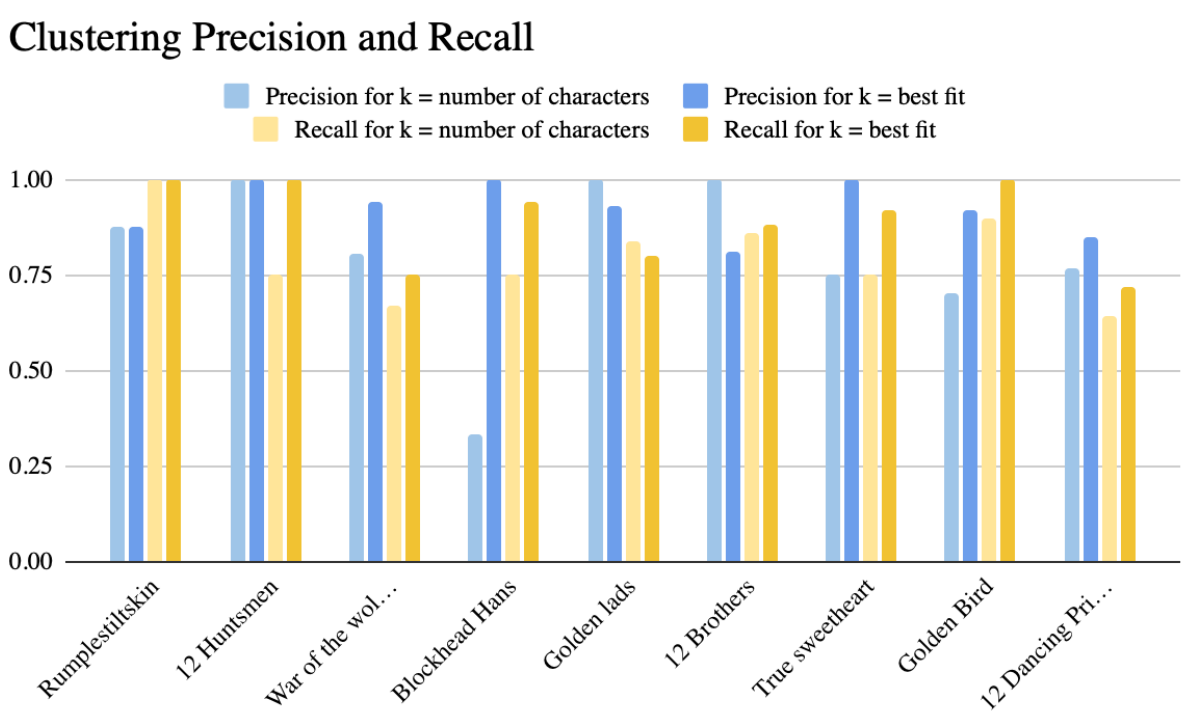

Figure 15: Clustering precision and recall for each document given k = number of characters and k = best fit

Thus with this approach to setting k there is much greater likelihood of a correct allocation of characters to groups, since it takes into account the potential for the incorrect inclusion of other nouns as main characters and the omission of a character potentially. The likelihood of a random extra noun that has been determined to be a character then is less likely to be grouped in with the actual character clusters simply because there are not enough clusters to account for an extra.

Overview of Results

With all the facets of the model in mind, let us look at two examples of the collection of characterizations. Given knowledge of the archetype of a character, we can see what commonalities may be found between their adjectives and verbs and those of other characters of the archetype, as well as how the two stories compare in terms of what archetypes appear. For the tale *Golden Mermaid*, we obtain the following:

King (Non-villainous family member)
ADJs: [['Most', 'able', 'golden', 'great', 'magic', 'old', 'other', 'young', 'youngest']]
VRBs: [['done', 'ended', 'had', 'hear', 'returned', 'said', 'stepped']]

Prince (Hero), guards (Miscellany) (mis-clustered)
ADJs: [['astonished', 'dear', 'delighted', 'despised', 'different', 'disappointed', 'fair', 'faithful', 'gloomy', 'golden', 'great', 'hungry', 'last', 'latter', 'lovely', 'mighty', 'more', 'outside', 'own', 'respectful', 'unfortunate', 'unhappy', 'watched', 'whole', 'young', 'youngest'], ['careful', 'respectful']]
VRBs: [['awaiting', 'biddin', 'begged', 'came', 'comforted', 'despised', 'ended', 'feel', 'felt', 'gone', 'got', 'had', 'happened', 'jumped', 'led', 'left', 'mounted', 'move', 'pretended', 'restrain', 'rode', 'seized', 'stood', 'stuck', 'told', 'took', 'turned'], ['aroused', 'awoke', 'ended', 'is', 'made']]

father (Non-villainous family member) (mis-clustered)
ADJs: [['disappointed', 'more', 'old', 'sad', 'shameful', 'youngest']]
VRBs: [['consented', 'ended', 'grew', 'hear']]

wolf (Helper)
ADJs: [['beautiful', 'exquisite', 'full', 'golden', 'good', 'grim', 'latter', 'lean-looking', 'least', 'lo', 'loud', 'many', 'mighty', 'more', 'most', 'outside', 'ship']]
VRBs: [['advised', 'bade', 'begged', 'comforted', 'ended', 'made', 'mounted', 'received', 'reminded', 'said', 'swung', 'told', 'turned']]

brothers (Villain)
ADJs: [['many', 'two']]
VRBs: [['despised', 'ended', 'made']]

Emperor (Dispatcher)
ADJs: [['disappointed', 'excellent', 'golden', 'guest', 'homeward', 'many', 'mighty', 'most', 'other', 'stable', 'valuable']]
VRBs: [['awoke', 'bent', 'brought', 'came', 'ended', 'helping', 'jumped', 'let', 'proceeded', 'promised', 'rode', 'ruled', 'said', 'sat', 'took', 'wearying']]

Court (not a main character)

ADJs: [['more']]
VRBs: [['ended', 'proceeded', 'reached', 'transformed']]

mermaid (Love interest)
ADJs: [['able', 'beautiful', 'dead', 'depressed', 'golden', 'grim', 'happy', 'little', 'most', 'much', 'poor']]
VRBs: [['acknowledge', 'bent', 'cheered', 'ended', 'gave', 'jump', 'lay', 'made', 'obtain', 'refused', 'sat', 'turned']]

And for the tale *Three Dwarves*, we obtain:

woman (Villain)
ADJs: [['Queen', 'bad', 'old', 'wicked']]
VRBs: [['made', 'rolled"]]

daughter, girl (Hero)
ADJs: [['beautiful', 'more', 'own', 'ugly'], ['beautiful', 'charming', 'little', 'obedient', 'own', 'poor']]
VRBs: [['came', 'did', 'drink', 'handing', 'hated', 'laid', 'rolled', 'standing', 'took', 'used', 'wash'], ['doing', 'hated', 'left', 'rolled', 'thought']]

water (not a character), Duck (Helper)
ADJs: [['pour'], []]
VRBs: [['pour', 'rolled', 'standing', 'wash'], ['rolled', 'said', 'swam', 'went']]

stepmother (Villain) (mis-clustered)
ADJs: [['furious', 'glad', 'old', 'ugly']]
VRBs: [['heard', 'jumped', 'rolled']]

Dwarfs, King (Love interest), men (Helpers) (King mis-clustered)
ADJs: [['little', 'snow'], ['bad', 'beautiful', 'ever.The', 'sorry'], ['little', 'three']]
VRBs: [['bid', 'rolled'], ['asked', 'came', 'come', 'drove', 'rolled', 'sat'], ['consulted', 'rolled']]

Queen (Hero) (misclustered)
ADJs: [[]]
VRBs: [['gave', 'rolled', 'took']]

Here we obtain two sets of characters and their associated descriptions. I have marked the actual archetype of each character, as well wrongly identified characters, wrongly identified adjectives and verbs, and mis-clustered character references. But more importantly let us consider commonalities that emerge between characters of a given archetype. I will particularly be drawing out types of adjectives or verbs used.[17] Only the non-villainous family member archetype is present in *Golden Mermaid*, but it is worth noting that his adjectives particularly define him as 'great,' similar to 'good,' which in Figure 1 (supplied again below) I presented as

[17] In a true automation of determining archetype based on descriptors, there would have to be a much larger collection of words across characters known to fit an archetype, of course, as well as likely the use of word similarity.

common in this archetype, and as ‘old,’ another common adjective for non-villainous family members (Figure 1). His verbs mostly relate to completion (‘done’ and ‘ended’) and speech (‘hear’ and ‘said’). There are notably no verbs that are very active (such as ‘run’ or ‘mount’), emotion-based (such as ‘cry’), or negatively toned (such as ‘hate’). Thus with this example we begin to see a certain picture emerge through the adjectives and verbs that are present and notably absent. It is evident how the adjectives used for the king tend to be in line with common adjectives of characters of this archetype, given the probabilities I have presented in Figure 1.

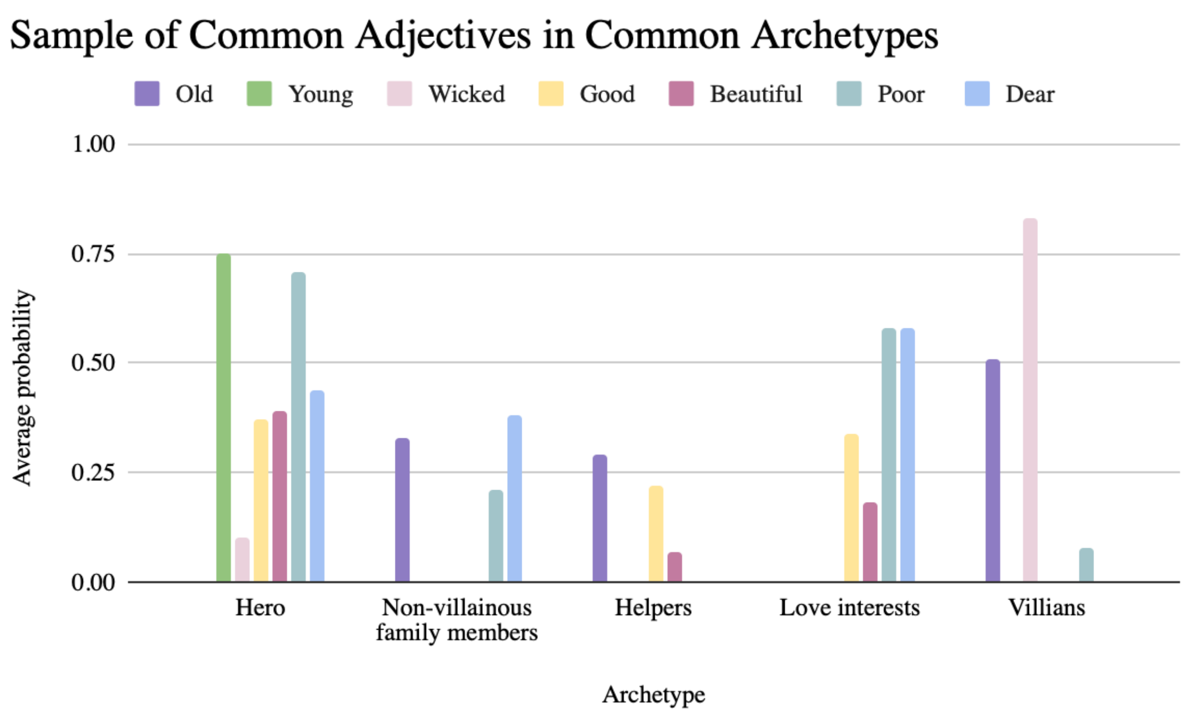

Figure 1: Average probability of each adjective in a given character archetype

This last example is somewhat simplified in terms of verbs, as it only tracks commonalities between the verbs used for a single character. For a fuller illustration of how verb usage functions, we should look at the archetypes of heroes and helpers, since we have multiple instances of these categories among the examples I have presented. In both *Golden Mermaid* and *Three Dwarves,* we have a hero (a prince and a girl), at least one helper (the wolf, the duck, and the dwarves), a love interest (the mermaid and the king), and a villain (the brothers and the stepmother). It is important to stress that such overlap in archetypes is not always the case; even a specific hero or a villain, what we might expect to be necessary staples of a fairy tale, are not both present in every story. But focusing on the descriptors, we find recurring types of adjectives suggesting ideal qualities for heroes, either in terms of good moral character (such as 'faithful,' 'respectful,' and 'obedient') or in terms of good looks (such as 'beautiful'). There are also similarities in words denoting an unfortunate state ('poor' describing the girl and 'unhappy' and 'unfortunate,' among others, describing the prince). With verbs heroes are involved in, there are particularly bodily actions ('standing,' 'moving,' and 'taking,' for example).

In contrast we may consider the helper characters. There are often fewer adjectives in this archetype, as we may note particularly in *Three Dwarves*, so verbs will be particularly helpful here. For the wolf, there are adjective types of goodness and beauty similar to heroes, but there is also 'mighty,' suggesting greater strength. Verbs, though, are more helpful in distinguishing this archetype, since here we find many words of speaking and of cognition ('consulted,' 'said,' and 'advised' are examples). There is a much higher concentration of such words in this class than among heroes: of the prince's 27 associated verbs, only three pertain to speaking and one is passive ('bidden') and one emotionally-tinged ('begged'). Thus a clear distinction between these groups appears here. While as I noted the non-villainous family member has more speaking verbs than the hero, he is differentiated by the balance of action words and the type of adjectives used. Given these factors, we are able to see commonalities not simply between individual words but between types of words. This enables us to group together characters into the same archetype and at the same time differentiate them from others of differing archetypes. The groupings of multiple similar adjectives and verbs for a given character better defines their archetype than simply the presence or absence of a single type of adjective or verb.

I will not go so closely through the other archetypes, but will note that villains are distinguishable particularly by adjectives denoting negative morals and often by a combination of speaking verbs and violently-tinged verbs in general. Love interests appear similarly to heroes often but generally have a higher number of looks-based descriptors and greater speaking verbs. Dispatchers are often described with adjectives relating to power (such as ‘mighty’ and ‘excellent’) and verbs relating to speaking.

Beyond simply categorizing the characters into archetypes, I argue that taking these archetypes together can suggest a type of story, which could be further concretized through a close consideration of Propp 1968’s allocations of specific actions to specific character types. Thus in the *The Golden Mermaid*, we may note how the presence of dispatcher suggests an overall quest that the hero undertakes. The presence of two helpers in *Three Dwarves* suggests a greater focus on a troubled hero being helped by multiple sources along the main, a common trope. Thus while the archetypes of course do not immediately tell us what specific type of story this is (this would need further research into tracking correlations between archetypes and types of events that occur), they do suggest a possible story arc.

Throughout this section I have underlined the varying accuracy of the individual components–character retrieval, adjective and verb retrieval, and clustering to address a character overcount. I have further explained how this work could be utilized to categorize each character into an archetype and the whole story into a specific type based on those archetypes. The work done in this project is focused on creating only a foundation for automated categorization systems, a foundation which is necessary for the task of finding correlations between types of adjectives and verbs for a given character and archetypes. With that in place, we have the capacity to actually calculate the probabilities which would be the basis for an automated assignment. This project is generally accurate, averaging F1 scores of 0.89, 0.75, and 0.66 on character retrieval, adjective extraction, and verb extraction, respectively, with an overall F1 score of 0.77. In addition, it displays coherent commonalities between members of different archetypes as I have drawn out in the overview of results. Because of these factors, I would count the project as successful.

Discussion

Given the mechanisms of the model and the results that I have cataloged, I will consider specific choices I have made in this project that affect its performance. Based on this, I will discuss limitations of this model and what that suggests for changing the model as currently configured. I will also consider the implications of my work here for future work into summarization and categorization of fairy tales, as well as the larger scope and importance of character-focused work such as this.

Limitations and Future Work

I will begin with a basic task that I chose to omit in my work here: that of coreference resolution. Coreference resolution attempts to identify pronouns with the nearby name they are most likely to refer to, thus allowing for better extraction of all mentions of a given entity. I decided to focus solely on the information that could be gleaned from specifically named references to the character, which I found to be effective overall, particularly in the fact that it did not lead to missing of main characters consistently or the like. Coreference resolution was outside of the scope of the project due to how incorrect I found the resolutions to be overall on fairy tales. This is likely due to the fact that these models are trained on datasets such as news articles, academic papers, and Wikipedia which diverge significantly in sentence structure from genres of fiction in general. Still most projects doing similar work on fairy tales or fiction–including but not limited to Adolfo & Ong 2019, Lovering 2016, R. Zhang et al. 2015, and W. Zhang et al. 2019–all utilize coreference resolution systems, in particular those from the Stanford CoreNLP Group. Thus adding coreference resolution would both be more in line with previous work and would more fully gather verbs and adjectives used for a given character, potentially aiding the potential omission of key, archetype-defining description.

A second choice important to consider is the use of Propp 1968 to define the basis of archetypes, as well as the related choice of how I have adapted his archetypes. I believe his work is a useful framing agent, serving to root the work in this project in a detailed, expert-derived analysis of folklore. It is also seems to me a good choice given that his work is the starting point for much NLP research into fairy tales (Droog-Hayes 2019, Declerck et al. 2010, Lendvai et al. 2010, Volkova et al. 2010, to remind us of a few I have previously mentioned). Yet an important caveat to note here is that of course Propp's analyses were performed on Russian folklore. While there are certainly coherent trends among different types of European folklore, and a number of NLP researchers have indeed extended Propp's work to other genres (El Maarouf et al. 2012, Volkova et al. 2010, and Declerck et al. 2010, among others), Russian folklore does have tropes specific to it. I have addressed this in my project to some extent by my proposed adaptations of Propp's character types to better fit the Grimm's tales I use. But for further work this would likely need to be more closely studied with more precise divisions devised, both with respect to the character types and with their adaptation to archetypal events. By automatically grouping a large collection of characters from different stories based on the similarity of their adjectives and verbs, an extension of the work I have done here could be used to help determine what archetypes are most suited to a given genre. This is then to say that there is a good deal of future work to be done to extend Propp's findings to genres other than Russian folklore.

Beyond this consideration of the framing via Propp, another key choice to examine is how I collected adjectives and the issues that arose from my choice. In my results, I highlighted the problem of the imprecision of the adjective retrieval system which ultimately I find to be a major defect of the model. While the collection of verbs could be considered as well, given the closeness of the two group's F1 scores, I found this category to lead to less abject confusion concerning a characterization. It overall struggled much more with recall than with precision, while with adjective collection it was the inverse, and the precision issues appear more troublesome. The adjective collection may introduce conflicting descriptors (such as 'young' and 'old') that point to different archetypes or the same adjective(s) being listed in most or all characterizations so that (at least on this basis) characters appear to all have the same archetype-defining information. In general even with large omissions of the total group of verbs associated with a character, the verb retrieval collects a relatively representative group, due to the often repetitious use of important actions (and descriptors) in fairy tales.

In my approach I decided to collect adjectives by targeting adjectives that appear right before a given character reference or that appear within a set distance past the character reference. This depends on the assumption that most adjectives that appear in that range will be in reference to the character they follow, thus concerning myself more with recall than precision. This assumption is a rather simplistic one, not utilizing machine learning approaches based on patterns of adjective appearance in sentences over a large corpus (though, while not a perfect comparison given our different datasets, I should note my approach was still more accurate than W. Zhang et al. 2019's collection utilizing Stanford's dependency parser). And the results likewise mirror the naiveté of such an approach, blindly extracting all words identified as adjectives. An obvious change or expansion of my work here would be to adjust the approach to adjective collection. W. Zhang et al. 2019 uses BookNLP and its dependency parser adjective modifier label, which is in line with most other researchers I have come across in using Stanford's NLP models. Thus while performance was better than what I found using the SpaCy dependency parser for this task, it was still not ideal, specifically lacking in its recall of the targeted adjectives. Another path might be, with all collections of adjectives, to see if outliers among each collection could be identified using clustering or word similarity methods. Words that are being listed in all character descriptions could also be targeted and omitted, though this could lead to the loss of important information for one or more characters, especially given the minimal adjectival description of some types of characters.

A final path (potentially the most ideal) might be to use a dependency parser specifically trained on a dataset of fairy tales rather than SpaCy's pretrained model I utilized. This could lead to better recognition of where adjective modifiers of characters generally appear in this specific genre. This final task suggests the additional research needed into the creation of data of fairy tales annotated for the correct adjectives of each character upon which to train the model, an area I have noted already as greatly important. While large corpuses exist for news articles and academic papers, despite the immense amount of fiction online there is very little of the same organization concerning this genre. For fairy tales, I have previously noted the important work of El Maarouf et al. 2012, Lobo & Matos 2010, Lendvai et al. 2010, and Declerck et al. 2010 on the problem of lack of annotated fairy tale corpora but, as they underline, the area is still underdeveloped. The corpus of Lobo & Matos 2010 is composed of only 453 texts, despite spanning European, Indian, Japanese, and Arabian fairy tales. El Maarouf et al. 2012 focuses on tales in French and Lendvai et al. 2010 on tales in German and these corpora are also both small in comparison with the usual NLP text collection. Thus, as underlined in this project and by the others, for advancements in natural language processing with fiction much work is needed on creating online corpora to utilize, corpora which should include gold standard summaries and text annotations.

Another area of discussion is my choices around clustering. I decided to cluster characterizations to group disparate character references and then descriptors into a single characterization. I utilized a pre-trained model of Glove for vector representations of the characterizations, which were created by averaging the vectors of each word within. This could perhaps be better approached, given the loss of information due to averaging. It does not take into account words that appear more frequently than others–thus, we might assume, more character-specific–and there is in general no weighting of types of words (name vs adjectives vs verbs) or specific words as more important than others, all of which could potentially lead to better results. There is also the improvement that could come from using a Glove model trained on fairy tales rather than the Wikipedia-trained one I utilized. Better vector representations could lead to better clustering of the characterizations. Besides vectorization, there is also the choice of type of clustering. I utilized k means but another type might be better suited, such as hierarchical clustering as it has an intuitive similarity to the act of categorization, building up larger and larger groupings of similar units in a bottom up way.

In terms of my collection technique, there is a clear aspect of the story that I have disregarded in my current approach to retrieval of adjectives and verbs: the fact that these words do not appear in isolation but in the context of sentences and an ongoing plot. I have taken them to have evident meaning and import based solely on their definition but the context a word appears in may lend it particular meaning and less or greater import to a given character. An apt example of this is that in the story *Snow White and Rose Red* at one point in the story the dwarf (the villain of sorts of the tale) calls the two girls 'wicked' in his anger at them. Without this context, 'wicked' appears alone as a descriptor of the heroines, suggesting very different characters than is appropriate and correct. For this type of case, greater attention could be paid to where the adjective occurs–namely if it is in the main narration or if it is in speech, the latter then requiring some evaluation of the relationship between speaker and the character being described.

There is also the factor of story arc in suggesting word importance. The first introduction of a character, often for the hero at least at the beginning of the story, will hold the most key attributes of the character, the adjectives which define them. For verbs, an attention to story arc might draw out the most important actions of a character, perhaps as defined by greatest effect on the course of the plot. For example, after the moment where the prince rescues the princess the general action of stories tend to change–before, events focus on journeying and facing obstacles whereas after the events such as marriage and generally celebrating will be the focus. Moreover the action of rescuing essentially defines him as the prince, as the hero at the end of his quest. This type of evaluation would require a much more semantically sensitive system implementing machine learning techniques in all likelihood, bringing forth questions of how to extract events and furthermore how a system might learn importance. The work of Adolfo & Ong 2019, Lovering 2016, and Goyal et al. 2010 would be a clear starting point for such an endeavor. We also might imagine that such attention to the context I have left out in this endeavor would yield the potential to extract the dynamic development of characters and plot as drawn out by knowledge of important events. This could lead to the ability to extract further significant themes.

Broader Implications

The consideration of dynamic changes in characters (perhaps even in the archetype they fall into) as defined by how descriptors shift in the course of the story suggests something important about the broader application of such work. Firstly this type of approach might be even more applicable to other types of fiction due to the generally static existence of characters in fairy tales. Granted, in fairy tales there are often shifts in social positions and perhaps in dynamics between characters but the topic would be most interesting in larger texts. We might imagine that groups of similar adjectives and verbs could be tracked in the course of a story to find points of divergence where similarity decreases; this would potentially suggest a change in the type of part a given character is playing. Likewise, pairs of characters could be compared at different points in a tale in respect to the verbal actions that associate them and considering for congruence or incongruence. From this we could gain not simply a static view of a given story but the more representative *evolution* throughout. This would be helpful for categorizing a story, fairy tale or not, with regard to the evolution a character does or does not undergo: we might be able to separate plot-driven narratives from those rooted in changes in personality and relationships and thereby define specific archetypal developments that may occur.

Besides extensions such as these in terms of task, I believe this project has important implications for application to other genres. It would be helpful for other genres of fiction, determining whether a piece of writing is more plot-driven or focused on character development, as I have suggested. Thus a group of characterizations of each main character gives a kind of sketch of the tale: while details of specific events may be lacking, we get a sense of the key players, what type of persons they are, and in what way they affect the action. This gives a mechanism to both understand a story by itself and to compare it with others. That applies when that other story may be another version of the same basic tale, as is often seen with folktales, with the characterizations giving a sense of how these versions differ. It also applies when we have two completely different stories, allowing us to gain an understanding of how they differ in category (whether they are character-driven or plot-driven, genre, etc.) and theme.

But we might also expand our purview beyond fiction altogether. Indeed, it's worth noting that there are similar "character" categorization efforts in other genres already. Supervised learning has been used for the categorization of entities as types of person (such as politician, actor, musician, etc.) based on the context in which they appear. Ren et al. 2016 seeks to create an approach that goes beyond a single context to more accurately define a person as a given type overall, given the often inaccurate assignment when it is based on a single instance. Going beyond identification of people as a given type, there is work on the identification of given short texts in terms of their political leaning (Maynard & Funk 2011). Maynard & Funk 2011 work specifically on short texts, specifically Tweets, and determine opinion based on the collection of a set of typical persons, their opinion, and the political party at hand. Mulholland & Quinn 2013 focus on vocabulary usage in a variety of songs by different lyricists to predict the likelihood, based on song lyrics, that a lyricist has or will try to commit suicide, thus similarly attempting to categorize individuals essentially based on word usage. With these examples, we see a trend in interest in classification either of texts based on the individuals that make it up and the word choice used or the classification of people based on the contexts in which they appear–trends that are similar to tracking the adjectives and verbs associated with a character and assigning him to a particular archetype. Thus my work seems like another approach that might be taken in tackling such questions.

To take a concrete example of a potential implication of usefulness, let us consider news reports. In an event discussed in a news report, there are agents of said event, types of actions they are performing, and relationships between the different agents. And yet there are also different news outlets which will portray an event in different ways generally, perhaps subtly or perhaps quite obviously. While these differences may be fully conflicting accounts of events, they are also likely to include different representations of the people involved, perhaps in terms of their actions or in terms of the ways in which they are described. Given this, we might see how such a characterization model could be highly useful in sorting through different news sources. We might imagine that using the information of how the same individuals are described in different news sources, utilizing clustering we could classify and even quantify the political leanings of articles given known themes in the representation of specific people across different political views. But beyond simply detecting negative or positive bias, such as with sentiment analysis, common themes might be found of what archetypes are used more so by different publications with different political views, giving a sense of the type of people they construe to be present in the world, the types of stories they tell generally. The same could be done with different types of political speech more broadly, potentially yielding useful correlations between the use of certain archetypes and types of stories and particular kinds of behavior. Imagining these sorts of applications of the model, we see how it could have great utility in a variety of areas beyond the scope of fictional texts.

Conclusion

My goal in this project has been to develop a type of summarization of fairy tales, specifically a method of categorization. This method involves extracting features of the main characters and their attributes in the form of adjectives and verbs. The main characters are identified as those nouns that appear frequently as verb subjects; adjectives are retrieved based on a set distance from the occurence of one of these identified characters; and verbs are extracted on the basis of a character being its subject or object. Due to a problem of character overcount that arose in a subset of the dataset, clustering with k means clustering was utilized as a means of associating references (and their associated descriptors) with the same character so as to accurately present the tale. While there are some issues with the model, particularly in respect to its overcollection of adjectives, it is successful in obtaining characterizations, averaging F1 scores of 0.89, 0.75, and 0.66 on character retrieval, adjective extraction, and verb extraction, respectively. Thus the model has an overall F1 score of 0.77.

The utility of these descriptions, or characterizations, is to present certain character archetypes. As I have discussed, in fairy tales there are particular adjectives which are standardly associated with particular types of characters–a pattern which holds consistently overall across the stories. Thus, villains have a much higher probability of being described as old and wicked than young or beautiful, whereas a hero or heroine is almost never described as old but has a high likelihood of being associated with youth and beauty. With a suggested method of categorization of the types of characters in a story, we are given a sense of the type of story at hand. Is there a villain? Or is there simply a hero and a love interest? Is a hero or heroine even present or is it simply lay people and helper side characters? While these elements alone do not describe a story arc, they are suggestive of what is or isn't possible and the likely plot. With the capacity to suggest character archetypes, the model I have designed thus provides a first step toward an overarching categorization of fairy tales.

The approach I have taken here can be extended and indeed has important broader implications. While an assignment to a given archetype has not been automatized in this project, the groundwork for such a mechanism has been laid and could fairly easily be developed. As I discussed in the previous section, a method similar to the proposed character archetype assignment could potentially be applied to the defining of event archetypes in order to develop a more robust categorization of story arc and thematic elements. Moreover, the approach taken here could eventually be applied far beyond the relatively narrow domain of fairy tales to other kinds of fiction and to non-fiction as well–scholarly books and papers, news articles, and so forth. Ultimately, the hope is to provide an automatized categorization schema, or perhaps multiple categorization schemata, for a broad range of content–a means of providing the user with some handle on the dizzying array of information that is now available on the internet. Needless to say, we are still a long way from being able to concretely realize such a possibility. But what I hope to have shown here is that this goal is not simply a fairy tale.

References

Adolfo, B. T., & Ong, E. (2019). Extracting Events from Fairy Tales for Story Summarization.

Agirrezabal, M., Altuna, B., Gil Vallejo, L., Goikoetxea, J., & Gonzalez Dios, I. (2018). Creating vocabulary exercises through NLP. Digital Humanities in the Nordic Countries. Proceedings, 2019.

Alm, C. O., & Sproat, R. (2005). Emotional sequencing and development in fairy tales. In International Conference on Affective Computing and Intelligent Interaction(pp. 668-674). Springer, Berlin, Heidelberg.

Bamman, D., O'Connor, B., & Smith, N. A. (2013). Learning latent personas of film characters. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 352-361).

Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.

Declerck, T., Scheidel, A., & Lendvai, P. (2010). Proppian Content Descriptors in an Augmented Annotation Schema for Fairy Tales. ECAI 2010, 35.

Droog-Hayes, M. (2019). Towards More Human-Like Text Summarization: Story Abstraction Using Discourse Structure and Semantic Information (Doctoral dissertation, Queen Mary University of London).

El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. Expert Systems with Applications, 165, 113679.

El Maarouf, I., & Villaneau, J. (2012). A French Fairy Tale Corpus syntactically and semantically annotated. In LREC'12(pp. 978-2).

Fernandez, M., Peterson, M., & Ulmer, B. (2015). Extracting social network from literature to predict antagonist and protagonist. Recuperado de: https://nlp. stanford. edu/courses/cs224n/2015/reports/14. pdf.

Goyal, A., Riloff, E., & Daumé III, H. (2010). Automatically producing plot unit representations for narrative text. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (pp. 77-86).

Goh, H. N., Soon, L. K., & Haw, S. C. (2012). Automatic identification of protagonist in fairy tales using verb. In Pacific-Asia Conference on Knowledge Discovery and Data Mining(pp. 395-406). Springer, Berlin, Heidelberg.

Grimm, J. & Grimm, W. (2001). *The Grimm's Fairy Tales.* Translated by Edgar Taylor and Marian Edwardes. Project Gutenberg. https://www.gutenberg.org/files/2591/2591-h/2591-h.htm

Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Kartika, L. G. S., Rinartha, K., & Siahaan, D. (2020). Text Summarization of Indonesian Folklore with Word Frequency Concept. In 2020 10th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS) (pp. 259-262). IEEE.

Lang, A. (2009). *The Fairy Books of Andrew Lang: A Project Gutenberg Linked Index to All Stories in the 12 Volumes.* Edited by David Widger. Project Gutenberg. https://www.gutenberg.org/files/30580/30580-h/30580-h.htm

Lendvai, P., Declerck, T., Darányi, S., Gervás, P., Hervás, R., Malec, S., & Peinado, F. (2010). Integration of Linguistic Markup into Semantic Models of Folk Narratives: The Fairy Tale Use Case. In: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10).

Lobo, P. V., & De Matos, D. M. (2010). Fairy tale corpus organization using latent semantic mapping and an item-to-item top-n recommendation algorithm. In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10).

Lovering, R. (2016). Event Parsing In Narrative: Trials And Tribulations Of Archaic English Fairy Tales.

Maynard, D., & Funk, A. (2011). Automatic detection of political opinions in tweets. In Extended semantic web conference (pp. 88-99). Springer, Berlin, Heidelberg.

Messina, L., Busso, L., Combei, C. R., Pannitto, L., Miaschi, A., Sarti, G., & Nissim, M. (2021). A dissemination workshop for introducing young Italian students to NLP. arXiv preprint arXiv:2104.12405.

Mulholland, M., & Quinn, J. (2013). Suicidal tendencies: The automatic classification of suicidal and non-suicidal lyricists using nlp. In Proceedings of the sixth international joint conference on natural language processing(pp. 680-684).

Nazari, N., & Mahdavi, M. A. (2019). A survey on automatic text summarization. Journal of AI and Data Mining, 7(1), 121-135.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Propp, V. J. (1968). Morphology of the folktale, University of Texas Press:, Austin.

Ren, X., He, W., Qu, M., Huang, L., Ji, H., & Han, J. (2016). Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding. In Proceedings of the 2016 conference on empirical methods in natural language processing (pp. 1369-1378).

Volkova, E. P., Mohler, B., Meurers, D., Gerdemann, D., & Bülthoff, H. H. (2010). Emotional perception of fairy tales: achieving agreement in emotion annotation of text. In Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text (pp. 98-106).

Zhang, W. (2017). Automatic summarization of fiction by generating character descriptions. McGill University (Canada).

Zhang, R., Li, W., Liu, N., & Gao, D. (2015). Coherent narrative summarization with a cognitive model. Computer Speech & Language, 35, 134-160.

Zhang, W., Cheung, J. C. K., & Oren, J. (2019). Generating character descriptions for automatic summarization of fiction. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 7476-7483).

Appendix

Golden Mermaid

A powerful king had, among many other treasures, a wonderful tree in his garden, which bore every year beautiful golden apples. But the King was never able to enjoy his treasure, for he might watch and guard them as he liked, as soon as they began to get ripe they were always stolen. At last, in despair, he sent for his three sons, and said to the two eldest, ‘Get yourselves ready for a journey. Take gold and silver with you, and a large retinue of servants, as beseems two noble princes, and go through the world till you find out who it is that steals my golden apples, and, if possible, bring the thief to me that I may punish him as he deserves.' His sons were delighted at this proposal, for they had long wished to see something of the world, so they got ready for their journey with all haste, bade their father farewell, and left the town.

The youngest Prince was much disappointed that he too was not sent out on his travels; but his father wouldn't hear of his going, for he had always been looked upon as the stupid one of the family, and the King was afraid of something happening to him. But the Prince begged and implored so long, that at last his father consented to let him go, and furnished him with gold and silver as he had done his brothers. But he gave him the most wretched horse in his stable, because the foolish youth hadn't asked for a better. So he too set out on his journey to secure the thief, amid the jeers and laughter of the whole court and town.

His path led him first through a wood, and he hadn't gone very far when he met a lean-looking wolf who stood still as he approached. The Prince asked him if he were hungry, and when the wolf said he was, he got down from his horse and said, ‘If you are really as you say and look, you may take my horse and eat it.'

The wolf didn't wait to have the offer repeated, but set to work, and soon made an end of the poor beast. When the Prince saw how different the wolf looked when he had finished his meal, he said to him, ‘Now, my friend, since you have eaten up my horse, and I have such a long way to go, that, with the best will in the world, I couldn't manage it on foot, the least you can do for me is to act as my horse and to take me on your back.'

‘Most certainly,' said the wolf, and, letting the Prince mount him, he trotted gaily through the wood. After they had gone a little way he turned round and asked his rider where he wanted to go to, and the Prince proceeded to tell him the whole story of the golden apples that had been stolen out of the King's garden, and how his other two brothers had set forth with many followers to find the thief. When he had finished his story, the wolf, who was in reality no wolf but a mighty magician, said he thought he could tell him who the thief was, and could help him to secure him. ‘There lives,' he said, ‘in a neighbouring country, a mighty emperor who has a beautiful golden bird in a cage, and this is the creature who steals the golden apples, but it flies so fast that it is impossible to catch it at its theft. You must slip into the Emperor's palace by night and steal the bird with the cage; but be very careful not to touch the walls as you go out.'

The following night the Prince stole into the Emperor's palace, and found the bird in its cage as the wolf had told him he would. He took hold of it carefully, but in spite of all his caution he touched the wall in trying to pass by some sleeping watchmen. They awoke at once, and, seizing him, beat him and put him into chains. Next day he was led before the Emperor, who at once condemned him to death and to be thrown into a dark dungeon till the day of his execution arrived.

The wolf, who, of course, knew by his magic arts all that had happened to the Prince, turned himself at once into a mighty monarch with a large train of followers, and proceeded to the Court of the Emperor, where he was received with every show of honour. The Emperor and he conversed on many subjects, and, among other things, the stranger asked his host if he had many slaves. The Emperor told him he had more than he knew what to do with, and that a new one had been captured that very night for trying to steal his magic bird, but that as he had already more than enough to feed and support, he was going to have this last captive hanged next morning.

‘He must have been a most daring thief,' said the King, ‘to try and steal the magic bird, for depend upon it the creature must have been well guarded. I would really like to see this bold rascal.' ‘By all means,' said the Emperor; and he himself led his guest down to the dungeon where the unfortunate Prince was kept prisoner. When the Emperor stepped out of the cell with the King, the latter turned to him and said, ‘Most mighty Emperor, I have been much disappointed. I had thought to find a powerful robber, and instead of that I have seen the most miserable creature I can imagine. Hanging is far too good for him. If I had to sentence him I should make him perform some very difficult task, under pain of death. If he did it so much the better for you, and if he didn't, matters would just be as they are now and he could still be hanged.' ‘Your counsel,' said the Emperor, ‘is excellent, and, as it happens, I've got the very thing for him to do. My nearest neighbour, who is also a mighty Emperor, possesses a golden horse which he guards most carefully. The prisoner shall be told to steal this horse and bring it to me.'

The Prince was then let out of his dungeon, and told his life would be spared if he succeeded in bringing the golden horse to the Emperor. He did not feel very elated at this announcement, for he did not know how in the world he was to set about the task, and he started on his way weeping bitterly, and wondering what had made him leave his father's house and kingdom. But before he had gone far his friend the wolf stood before him and said, ‘Dear Prince, why are you so cast down? It is true you didn't succeed in catching the bird; but don't let that discourage you, for this time you will be all the more careful, and will doubtless catch the horse.' With these and like words the wolf comforted the Prince, and warned him specially not to touch the wall or let the horse touch it as he led it out, or he would fail in the same way as he had done with the bird.

After a somewhat lengthy journey the Prince and the wolf came to the kingdom ruled over by the Emperor who possessed the golden horse. One evening late they reached the capital, and the wolf advised the Prince to set to work at once, before their presence in the city had aroused the watchfulness of the guards. They slipped unnoticed into the Emperor's stables and into the very place where there were the most guards, for there the wolf rightly surmised they would find the horse. When they came to a certain inner door the wolf told the Prince to remain outside, while he went in. In a short time he returned and said, ‘My dear Prince, the horse is most securely watched, but I have bewitched all the guards, and if you will only be careful not to touch the wall yourself, or let the horse touch it as you go out, there is no danger and the game is yours. The Prince, who had made up his mind to be more than cautious this time, went cheerfully to work. He found all the guards fast asleep, and, slipping into the horse's stall, he seized it by the bridle and led it out; but, unfortunately, before they had got quite clear of the stables a gadfly stung the horse and caused it to switch its tail, whereby it touched the wall. In a moment all the guards awoke, seized the Prince and beat him mercilessly with their horse-whips, after which they bound him with chains, and flung him into a dungeon. Next morning they brought him before the Emperor, who treated him exactly as the King with the golden bird had done, and commanded him to be beheaded on the following day.

When the wolf-magician saw that the Prince had failed this time too, he transformed himself again into a mighty king, and proceeded with an even more gorgeous retinue than the first time to the Court of the Emperor. He was courteously received and entertained, and once more after dinner he led the conversation on to the subject of slaves, and in the course of it again requested to be allowed to see the bold robber who had dared to break into the Emperor's stable to steal his most valuable possession. The Emperor consented, and all happened exactly as it had done at the court of the Emperor with the golden bird; the prisoner's life was to be spared only on condition that within three days he should obtain possession of the golden mermaid, whom hitherto no mortal had ever approached.

Very depressed by his dangerous and difficult task, the Prince left his gloomy prison; but, to his great joy, he met his friend the wolf before he had gone many miles on his journey. The cunning creature pretended he knew nothing of what had happened to the Prince, and asked him how he had fared with the horse. The Prince told him all about his misadventure, and the condition on which the Emperor had promised to spare his life. Then the wolf reminded him that he had twice got him out of prison, and that if he would only trust in him, and do exactly as he told him, he would certainly succeed in this last undertaking. Thereupon they bent their steps towards the sea, which stretched out before them, as far as their eyes could see, all the waves dancing and glittering in the bright sunshine. ‘Now,' continued the wolf, ‘I am going to turn myself into a boat full of the most beautiful silken merchandise, and you must jump boldly into the boat, and steer with my tail in your hand right out into the open sea. You will soon come upon the golden mermaid. Whatever you do, don't follow her if she calls you, but on the contrary say to her, "The buyer comes to the seller, not the seller to the buyer." After which you must steer towards the land, and she will follow you, for she won't be able to resist the beautiful wares you have on board your ship.'

The Prince promised faithfully to do all he had been told, whereupon the wolf changed himself into a ship full of most exquisite silks, of every shade and colour imaginable. The astonished Prince stepped into the boat, and, holding the wolf's tail in his hand, he steered boldly out into the open sea, where the sun was gilding the blue waves with its golden rays. Soon he saw the golden mermaid swimming near the ship, beckoning and calling to him to follow her; but, mindful of the wolf's warning, he told her in a loud voice that if she wished to buy anything she must come to him. With these words he turned his magic ship round and steered back towards the land. The mermaid called out to him to stand still, but he refused to listen to her and never paused till he reached the sand of the shore. Here he stopped and waited for the mermaid, who had swum after him. When she drew near the boat he saw that she was far more beautiful than any mortal he had ever beheld. She swam round the ship for some time, and then swung herself gracefully on board, in order to examine the beautiful silken stuffs more closely. Then the Prince seized her in his arms, and kissing her tenderly on the cheeks and lips, he told her she was his for ever; at the same moment the boat turned into a wolf again, which so terrified the mermaid that she clung to the Prince for protection.

So the golden mermaid was successfully caught, and she soon felt quite happy in her new life when she saw she had nothing to fear either from the Prince or the wolf—she rode on the back of the latter, and the Prince rode behind her. When they reached the country ruled over by the Emperor with the golden horse, the Prince jumped down, and, helping the mermaid to alight, he led her before the Emperor. At the sight of the beautiful mermaid and of the grim wolf, who stuck close to the Prince this time, the guards all made respectful obeisance, and soon the three stood before his Imperial Majesty. When the Emperor heard from the Prince how he had gained possession of his fair prize, he at once recognized that he had been helped by some magic art, and on the spot gave up all claim to the beautiful mermaid. ‘Dear youth,' he said, ‘forgive me for my shameful conduct to you, and, as a sign that you pardon me, accept the golden horse as a present. I acknowledge your power to be greater even than I can understand, for you have succeeded in gaining possession of the golden mermaid, whom hitherto no mortal has ever been able to approach.' Then they all sat down to a huge feast, and the Prince had to relate his adventures all over again, to the wonder and astonishment of the whole company.

But the Prince was wearying now to return to his own kingdom, so as soon as the feast was over he took farewell of the Emperor, and set out on his homeward way. He lifted the mermaid on to the golden horse, and swung himself up behind her—and so they rode on merrily, with the wolf trotting behind, till they came to the country of the Emperor with the golden bird. The renown of the Prince and his adventure had gone before him, and the Emperor sat on his throne awaiting the arrival of the Prince and his companions. When the three rode into the courtyard of the palace, they were surprised and delighted to find everything festively illuminated and decorated for their reception. When the Prince and the golden mermaid, with the wolf behind them, mounted the steps of the palace, the Emperor came forward to meet them, and led them to the throne room. At the same moment a servant appeared with the golden bird in its golden cage, and the Emperor begged the Prince to accept it with his love, and to forgive him the indignity he had suffered at his hands. Then the Emperor bent low before the beautiful mermaid, and, offering her his arm, he led her into dinner, closely followed by the Prince and her friend the wolf; the latter seating himself at table, not the least embarrassed that no one had invited him to do so.

As soon as the sumptuous meal was over, the Prince and his mermaid took leave of the Emperor, and, seating themselves on the golden horse, continued their homeward journey. On the way the wolf turned to the Prince and said, 'Dear friends, I must now bid you farewell, but I leave you under such happy circumstances that I cannot feel our parting to be a sad one.' The Prince was very unhappy when he heard these words, and begged the wolf to stay with them always; but this the good creature refused to do, though he thanked the Prince kindly for his invitation, and called out as he disappeared into the thicket, 'Should any evil befall you, dear Prince, at any time, you may rely on my friendship and gratitude.' These were the wolf's parting words, and the Prince could not restrain his tears when he saw his friend vanishing in the distance; but one glance at his beloved mermaid soon cheered him up again, and they continued on their journey merrily.

The news of his son's adventures had already reached his father's Court, and everyone was more than astonished at the success of the once despised Prince. His elder brothers, who had in vain gone in pursuit of the thief of the golden apples, were furious over their younger brother's good fortune, and plotted and planned how they were to kill him. They hid themselves in the wood through which the Prince had to pass on his way to the palace, and there fell on him, and, having beaten him to death, they carried off the golden horse and the golden bird. But nothing they could do would persuade the golden mermaid to go with them or move from the spot, for ever since she had left the sea, she had so attached herself to her Prince that she asked nothing else than to live or die with him.

For many weeks the poor mermaid sat and watched over the dead body of her lover, weeping salt tears over his loss, when suddenly one day their old friend the wolf appeared and said, 'Cover the Prince's body with all the leaves and flowers you can find in the wood.' The maiden did as he told her, and then the wolf breathed over the flowery grave, and, lo and behold! the Prince lay there sleeping as peacefully as a child. 'Now you may wake him if you like,' said the wolf, and the mermaid bent over him and gently kissed the wounds his brothers had made on his forehead, and the Prince awoke, and you may imagine how delighted he was to find his beautiful mermaid beside him, though he felt a little depressed when he thought of the loss of the golden bird and the golden horse. After a time the wolf, who had likewise fallen on the Prince's neck, advised them to continue their journey, and once more the Prince and his lovely bride mounted on the faithful beast's back.

The King's joy was great when he embraced his youngest son, for he had long since despaired of his return. He received the wolf and the beautiful golden mermaid most cordially too, and the Prince was made to tell his adventures all over from the beginning. The poor old father grew very sad when he heard of the shameful conduct of his elder sons, and had them called before him. They turned as white as death when they saw their brother, whom they thought they had murdered, standing beside them alive and well, and so startled were they that when the King asked them why they had behaved so wickedly to their brother they could think of no lie, but confessed at once that they had slain the young Prince in order to obtain possession of the golden horse and the golden bird. Their father's wrath knew no bounds, and he ordered them both to be banished, but he could not do enough to honour his youngest son, and his marriage with the beautiful mermaid was celebrated with much pomp and magnificence. When the festivities were over, the wolf bade them all farewell, and returned once more to his life in the woods, much to the regret of the old King and the young Prince and his bride.
And so ended the adventures of the Prince with his friend the wolf.

Three Dwarves

There was once upon a time a man who lost his wife, and a woman who lost her husband; and the man had a daughter and so had the woman. The two girls were great friends and used often to play together. One day the woman turned to the man's daughter and said:
‘Go and tell your father that I will marry him, and then you shall wash in milk and drink wine, but my own daughter shall wash in water and drink it too.'
The girl went straight home and told her father what the woman had said.
‘What am I to do?' he answered. ‘Marriage is either a success or it is a failure.'
At last, being of an undecided character and not being able to make up his mind, he took off his boot, and handing it to his daughter, said:
‘Take this boot which has a hole in the sole, hang it up on a nail in the hayloft, and pour water into it. If it holds water I will marry again, but if it doesn't I won't.' The girl did as she was bid, but the water drew the hole together and the boot filled up to the very top. So she went and told her father the result. He got up and went to see for himself, and when he saw that it was true and no mistake, he accepted his fate, proposed to the widow, and they were married at once.
On the morning after the wedding, when the two girls awoke, milk was standing for the man's daughter to wash in and wine for her to drink; but for the woman's daughter, only water to wash in and only water to drink. On the second morning, water to wash in and water to drink was standing for the man's daughter as well. And on the third morning, water to wash in and water to drink was standing for the man's daughter, and milk to wash in and wine to drink for the woman's daughter; and so it continued ever after. The woman hated her stepdaughter from the bottom of her heart, and did all she could to make her life miserable. She was as jealous as she could possibly be, because the girl was so beautiful and charming, while her own daughter was both ugly and repulsive.
One winter's day when there was a hard frost, and mountain and valley were covered with snow, the woman made a dress of paper, and calling the girl to her said:
‘There, put on this dress and go out into the wood and fetch me a basket of strawberries!'

‘Now Heaven help us,' replied her stepdaughter; ‘strawberries don't grow in winter; the earth is all frozen and the snow has covered up everything; and why send me in a paper dress? it is so cold outside that one's very breath freezes; the wind will whistle through my dress, and the brambles tear it from my body.'

‘How dare you contradict me!' said her stepmother; ‘be off with you at once, and don't show your face again till you have filled the basket with strawberries.'

Then she gave her a hard crust of bread, saying:

‘That will be enough for you to-day,' and she thought to herself: ‘The girl will certainly perish of hunger and cold outside, and I shan't be bothered with her any more.'

The girl was so obedient that she put on the paper dress and set out with her little basket. There was nothing but snow far and near, and not a green blade of grass to be seen anywhere. When she came to the wood she saw a little house, and out of it peeped three little dwarfs. She wished them good-day, and knocked modestly at the door. They called out to her to enter, so she stepped in and sat down on a seat by the fire, wishing to warm herself and eat her breakfast. The Dwarfs said at once: ‘Give us some of your food!'

‘Gladly,' she said, and breaking her crust in two, she gave them the half.

Then they asked her what she was doing in the depths of winter in her thin dress.

‘Oh,' she answered, ‘I have been sent to get a basketful of strawberries, and I daren't show my face again at home till I bring them with me.'

When she had finished her bread they gave her a broom and told her to sweep away the snow from the back door. As soon as she left the room to do so, the three little men consulted what they should give her as a reward for being so sweet and good, and for sharing her last crust with them.

The first said: ‘Every day she shall grow prettier.'

The second: ‘Every time she opens her mouth a piece of gold shall fall out.'

And the third: ‘A King shall come and marry her.'

The girl in the meantime was doing as the Dwarfs had bidden her, and was sweeping the snow away from the back door, and what do you think she found there?—heaps of fine ripe strawberries that showed out dark red against the white snow. She joyfully picked enough to fill her basket, thanked the little men for their kindness, shook hands with them, and ran home to bring her stepmother what she had asked for. When she walked in and said; Good evening,' a piece of gold fell out of her mouth. Then she told what had happened to her in the wood, and at every word pieces of gold dropped from her mouth, so that the room was soon covered with them.

‘She's surely more money than wit to throw gold about like that,' said her stepsister, but in her secret heart she was very jealous, and determined that she too would go to the wood and look for strawberries. But her mother refused to let her go, saying:

‘My dear child, it is far too cold; you might freeze to death.'

The girl however left her no peace, so she was forced at last to give in, but she insisted on her putting on a beautiful fur cloak, and she gave her bread and butter and cakes to eat on the way.

The girl went straight to the little house in the wood, and as before the three little men were looking out of the window. She took no notice of them, and without as much as ‘By your leave,' or ‘With your leave,' she flounced into the room, sat herself down at the fire, and began to eat her bread and butter and cakes.

‘Give us some,' cried the Dwarfs.

But she answered: ‘No, I won't, it's hardly enough for myself; so catch me giving you any.'

When she had finished eating they said:

‘There's a broom for you, go and clear up our back door.'
‘I'll see myself further,' she answered rudely. ‘Do it yourselves; I'm not your servant.'
When she saw that they did not mean to give her anything, she left the house in no amiable frame of mind. Then the three little men consulted what they should do to her, because she was so bad and had such an evil, covetous heart, that she grudged everybody their good fortune.
The first said: ‘She shall grow uglier every day.'
The second: ‘Every time she speaks a toad shall jump out of her mouth.'
And the third: ‘She shall die a most miserable death.'
The girl searched for strawberries, but she found none, and returned home in a very bad temper. When she opened her mouth to tell her mother what had befallen her in the wood, a toad jumped out, so that everyone was quite disgusted with her.
Then the stepmother was more furious than ever, and did nothing but plot mischief against the man's daughter, who was daily growing more and more beautiful. At last, one day the wicked woman took a large pot, put it on the fire and boiled some yarn in it. When it was well scalded she hung it round the poor girl's shoulder, and giving her an axe, she bade her break a hole in the frozen river, and rinse the yarn in it. Her stepdaughter obeyed as usual, and went and broke a hole in the ice. When she was in the act of wringing out the yarn a magnificent carriage passed, and the King sat inside. The carriage stood still, and the King asked her:
‘My child, who are you, and what in the wide world are you doing here?'
‘I am only a poor girl,' she answered, ‘and am rinsing out my yarn in the river.' Then the King was sorry for her, and when he saw how beautiful she was he said:
‘Will you come away with me?'
‘Most gladly,' she replied, for she knew how willingly she would leave her stepmother and sister, and how glad they would be to be rid of her.
So she stepped into the carriage and drove away with the King, and when they reached his palace the wedding was celebrated with much splendour. So all turned out just as the three little Dwarfs had said.
After a year the Queen gave birth to a little son. When her stepmother heard of her good fortune she came to the palace with her daughter by way of paying a call, and took up her abode there. Now one day, when the King was out and nobody else near, the bad woman took the Queen by her head, and the daughter took her by her heels, and they dragged her from her bed, and flung her out of the window into the stream which flowed beneath it. Then the stepmother laid her ugly daughter in the Queen's place, and covered her up with the clothes, so that nothing of her was seen. When the King came home and wished to speak to his wife the woman called out:
‘Quietly, quietly! this will never do; your wife is very ill, you must let her rest all to-day.' The King suspected no evil, and didn't come again till next morning. When he spoke to his wife and she answered him, instead of the usual piece of gold a toad jumped out of her mouth. Then he asked what it meant, and the old woman told him it was nothing but weakness, and that she would soon be all right again.
But that same evening the scullion noticed a duck swimming up the gutter, saying as it passed:
‘What does the King, I pray you tell,Is he awake or sleeps he well?'
and receiving no reply, it continued:
‘And all my guests, are they asleep?'
and the Scullion answered:
‘Yes, one and all they slumber deep.'

Then the Duck went on:
‘And what about my baby dear?'
and he answered:
‘Oh, it sleeps soundly, never fear.'
Then the Duck assumed the Queen's shape, went up to the child's room, tucked him up comfortably in his cradle, and then swam back down the gutter again, in the likeness of a Duck. This was repeated for two nights, and on the third the Duck said to the Scullion:
‘Go and tell the King to swing his sword three times over me on the threshold.'
The Scullion did as the creature bade him, and the King came with his sword and swung it three times over the bird, and lo and behold! his wife stood before him once more, alive, and as blooming as ever.
The King rejoiced greatly, but he kept the Queen in hiding till the Sunday on which the child was to be christened. After the christening he said:
‘What punishment does that person deserve who drags another out of bed, and throws him or her, as the case may be, into the water?'
Then the wicked old stepmother answered:
‘No better fate than to be put into a barrel lined with sharp nails, and to be rolled in it down the hill into the water.'
‘You have pronounced your own doom,' said the King; and he ordered a barrel to be made lined with sharp nails, and in it he put the bad old woman and her daughter. Then it was fastened down securely, and the barrel was rolled down the hill till it fell into the river.