Detection of Characters in Folktales

By
Unaiza Faiz
(UIN: 651052450)

Advisor: Prof. Barbara Di Eugenio
Committee: Prof. Natalie Parde

Project Report Submitted in partial fulfillment of requirements
for the degree of
Master of Science in Computer Science (Spring 2019)
at University of Illinois at Chicago

Table of Contents

Acknowledgment

I would like to extend my sincere thanks to all the individuals and organizations without whose kind support this project would not have been possible.

I am highly indebted to Prof. Barbara di Eugenio for their guidance and constant supervision throughout the development of this project. She steered me in the right direction during the research for the project, and also provided necessary reference and information when necessary which helped me during the course of this project.

I am grateful to Prof. Natalie Parde for being part of this project Committee. Her willingness to give her time so generously has been very much appreciated.

I am thankful for being part of UIC NLP lab. The bi-weekly meetings helped me understand research better and get the support from the lab members when needed, to whom I am greatly thankful for.

Lastly, I would like to thank my parents, whose love and guidance pushes me to do better and achieve greater heights in whatever I pursue.

Abstract

Detection of characters in folktales is a complex task, as these stories involve animals, objects etc., as characters. Characters are often identified with noun phrases like ‘the old woman’ in addition to proper nouns. Such character names are not detected by a Named Entity Recognizer. This project aims at detection of all characters in folktales using three methods. First, using Stanford CoreNLP parser for named entity recognition. Second, fine tuning the detection of all proper nouns using POS tagger and filtering using WordNet. Finally, characters identified by common nouns and/or noun phrases are extracted by identifying subject-verb and verb-object phrases. These phrases are filtered using two features of WordNet, namely sentence frame and derivationally related form to detect the verbs that are often associated with human activity. The characters detected from all methods are then combined to produce the final output and the results are evaluated.

1 Introduction

Named-entity recognition (NER) is a subtask in information extraction that seeks to locate and classify named entity mentions in unstructuctured text into predefined categories such as person names, location, organization etc.[1] However, a lot of NER systems are trained on non-fiction, due to which they do not perform well while working on a fiction corpus.

Fiction narratives involve several challenges. First, a "person" in the fiction domain could be either a human, animal or a non-living object. For example, in the story of Cinderella, teapots could be considered as "person" since they behave and exhibit characters of a human, like talking and moving.

Second, a person name is not necessarily a proper noun. Characters in the fiction domain are often identified in various ways like by a common noun (fairy) or using a qualifier that describes the nature a character exhibits with a common noun (the cruel stepmother). Such names often go unrecognized by an NER.

In this project, we propose to implement a system that detects characters in fiction narratives, specifically in children's folktales. Character can be defined as any person, animal, or figure represented in a literary work.[2] The proposed system aims at detection and extraction of these character names from a folktale.

A folktale is different from a fairytale. A folktale is a story that has originated with oral traditions that have been passed through generations. A fairytale may also have the same origins but additionally involves mythical characters like fairies, witches etc.[3]

One of the challenges of working on folktale data is the lack of annotated corpora in this domain. As a result, data collection and annotation were a major part of this project. A machine learning

[1] Wikipedia
[2] https://study.com/academy/lesson/character-in-literature-definition-types-development.html
[3] Difference between folktale and fairytale

approach could not be employed for this problem as that requires a large amount of annotated data to train the model.

Data was collected from the L^2F Fairy tale[4] corpus. Five stories from six different authors were selected to build the corpus. 18 stories were selected for training set, 6 for development set and 6 for test set. The stories were annotated by humans to identify characters and this annotated data was used as a gold standard while building and evaluating the system.

The proposed system uses three methods for this purpose. First, Stanford CoreNLP NER is used to extract all words tagged as "PERSON", this is also considered as the baseline. Second, all proper nouns are detected and filtered using WordNet. Finally, a variation of Goh et al.'s [6] system is used to identify human activity verbs in subject-verb and verb-object clauses and then use the corresponding noun to extract the characters in the story. Results from all the three are combined to produce a list of characters in the folktale.

The system was evaluated using the human annotated data as gold standard. Precision, recall, F-measure, and kappa were calculated for the system using two units. First, evaluation was done based on how many noun phrases in a story contain one of the characters identified by the model. Second, nouns were used as the unit of evaluation to calculate the number of nouns in the story (e.g. The noun *woman* in the character name "*an old woman"*) that are labelled as character by the system.

[4] https://www.l2f.inesc-id.pt/w/Fairy_tale_corpus

2 Related work

Several papers on named entity recognition in folktales and fairytales focus on character identification either as the main aim or part of a proposed system.

Gupta et al. [1] focus on extraction of character info boxes from books. For this task they parse the book text using POS Tagger and NER to identify person, place and organizations that occur. Person names are sorted by frequency of occurrence in the book and names less than a frequency threshold are removed.

Karsdorp et al. [2] propose a system that extracts ranked list of actors from fairytales (Dutch folktales) ordered by importance. The system defines character in folktales as 'actors' that express intentionality or consciousness. The paper shows that direct speech and indirect speech are effective indicators of intentionality and extracting these constructions from a text helps to retrieve the actors that form the cast of the story.

Suciu and Groza [3] propose a system for identifying characters in folktales using ontology that encodes knowledge of folktales. This approach does not identify some main characters such as Swan-Geese and Henry, because the system does not know how to treat characters that are specified by name.

Goh et al. [4] propose automatic protagonist identification in the fairytale domain. The verb is used as a determinant in identifying the existence of protagonist particularly in the "people" category with assistance of WordNet. Though this method uses both syntactic features and semantic lexicon for this purpose, the method is not extended to the rest of the actors.

Yeung and Lee [5] in their work propose a system that uses quoted speech to identify speaker and listener. This approach may not be extended to fairytales as we cannot assume that all fairytales contain direct speech sentences that are used in this method to identify the speakers.

Finally, Goh et all [6] in a different paper propose identification of verbs associated with human activity and then identify characters based on these verbs. Although in this paper the evaluation is done only on a short corpus of 7 popular fairytales, in our project we use a similar approach as described in this paper to identify human activity verbs.

The approaches used in all the previous works are evaluated either on a very small corpus of less than 10 stories that are commonly told like Rapunzel, Cinderella or the system is evaluated on only a specific domain of folktales, e.g. Grimm's tales, Dutch folktales. In contrast, our proposed project aims at building a system for folktales written by six different authors including some translated from other languages. We aim at generalizing the detection of characters in fairytales by using a broader spectrum of folktales as our dataset. The system aims at detecting not only named characters, but also nominal phrases used to identify these characters.

3 Dataset

The L^2F Fairy tale Corpus was used in this project [7], 30 of the 453 stories were considered for the corpus. The stories arbitrarily selected and equally distributed are written by six different authors, namely: Arabian writer, Beatrix Potter, Grimm's fairytales, H. C. Anderson, La Fountain and Indian writer. (The tales by H. C. Anderson are originally Danish fables and those by La Fountain are French and have been translated to English.)

18 stories were arbitrarily selected for training set, 3 stories from each author. 6 stories each were added to development and test corpus. The total number of words across the corpus is 107099, average number of sentences is 133 and average number of words per story is 3569.

3.1 Data Preprocessing

The data in its original form contained POS tags after each word and some special characters. All POS tags and special characters were deleted from the files, extracting only the words that make up the text of the story.

3.2 Data Annotation

The biggest challenge in working on a folktale corpus is the lack of annotated data. We thus had to use two human annotators to annotate the stories and identify the characters in the story. The annotators annotated 11 and 21 stories respectively. This annotation was then considered as the gold standard for evaluation.

The task of the annotators was to identify the characters in each story and note them in a list format in the corresponding text file of the story. The annotators were instructed to select noun phrases of the form <DET NN> (e.g. *the bird*), <DET JJ NN> (e.g. *the old king)*, <NN> (e.g. *merchant*), <NNP> (e.g. *Princess Bedoura*) to identify the characters (where DET is determiner, NN is common noun, JJ is adjective and NNP is proper noun).

Also, if a character is identified by multiple combinations of the same name, for instance, *Sir Isaac Newton / Isaac*, then the longest string is used to identify the character. If different names are used to identify the same character, then only the first name used is considered.

Inter coder agreement was calculated between the two annotators. Noun Phrases (NP) were extracted from the story using Stanford CoreNLP ParserAnnotator [5]. Each noun phrase was then marked as containing a character name (*the old king*) or not containing a character name based on the annotations of the two annotators. The confusion matrix obtained using these results is shown in Table 1. A Kappa value of 0.6201 was found.

It was observed from the annotations that each annotator had different views regarding when a common noun was used as a character. While one annotator considered a bird in the scene as a character, another annotator did not see this a character in the scene to be given preference to.

Table 1. Confusion matrix of annotation

		Coder 2		
		Character	Not a character	
Coder 1	Character	272	14	286
	Not a character	226	1359	1585
		498	1373	1871

[5] https://stanfordnlp.github.io/CoreNLP/parse.html

4 Methods

Three methods were used to retrieve the characters in a story.

4.1 Identify characters using the Stanford CoreNLP NER

Stanford CoreNLP Named-Entity Recognizer is used to tag entities with person, location, organization. In our project, we use the words tagged as PERSON to identify characters.

```
When Benjamin Bunny grew up, he married his Cousin Flopsy
      PERSON PERSON                           PERSON PERSON
```

To build character names from these tags a while loop logic is used. The output of this method is also considered as the baseline.

4.2 Identify characters based on NNP POS Tags

The challenge with using the Stanford CoreNLP NER is that it does not detect names with titles as full names. For instance, in the story of "The tale of Flopsy bunnies" by Beatrix Potter in our corpus, the characters are referred to as *Mr. McGregor* and *Mrs. McGregor*. The Stanford CoreNLP NER tagger only tags *McGregor* as PERSON, leading to the problem that the two unique characters are now identified as just one.

To take care of this problem we use POS Tags because the POS tagger tags all the words in the name as a proper noun (NNP). For instance,

```
Mr.  McGregor   tied   up   the   sack.
NNP     NNP     VBD    RP    DT    NN
```

One problem that was faced in this method is that it identifies words such as *Everybody* as proper nouns. So, such words were filtered from the list by finding if the word is present in the dictionary as an animate being using WordNet.

4.3 Identify characters by extracting human activity verbs from subject-verb and verb-object clauses

Goh et al. [6] propose a system that identifies verbs that are associated with human activity and detects characters in fairytales based on these verbs. We extend this system with some modification to detect the characters in our folktales.

The typed dependency parser from Stanford Core NLP is used to retrieve the following dependencies[6] for this method:

nn: *noun compound modifier*
A noun compound modifier of an NP is any noun that serves to modify the head noun.
e.g. For the clause, "*Oil price futures"* we get the dependencies *nn*(futures, oil) and *nn*(futures,price)

nsubj: *nominal subject*
A nominal subject is a noun phrase which is the syntactic subject of a clause. The governor of this subject might not be a verb, but in our system, we select only those values where verb is the governor. e.g. For the clause, *"Princess cried"* we get the dependency *nsubj* (cried, Prince)

nsubjpass: *passive nominal subject*
A passive nominal subject is a noun phrase which is the syntactic subject of a passive clause.
e.g. For the clause "*The king was defeated*" we get the dependency *nsubjpass* (defeated, king)

dobj: *direct object*
The direct object of a verb phrase is the noun phrase which is the (accusative) object of the verb.
e.g. For the clause "*the king praised the prince*" we get the dependency *dobj* (praised, prince)

amod: *adjective modifier*
An adjectival modifier of an noun phrase is any adjectival phrase that serves to modify the

[6] https://nlp.stanford.edu/software/dependencies_manual.pdf

meaning of the noun phrase.
e.g. For the clause "*an old woman*" we get the dependency *amod* (woman, old)

compound: *compound*
The compound relation is used to identify multiword expressions
e.g. For "*Isaac Newton*" we get the dependency *compound* (Newton, Isaac)

det: *determiner*
The relation determiner holds between a nominal head and its determiner.
e.g. For "the princess" we get the dependency *det* (princess, the)

In this method the first step is to extract a list of subject-verb(noun-verb) and verb-object(verb-noun) clauses using the typed dependencies nsubj, nsubjpass, nn, and dobj.

Subject-verb (noun-verb) clauses like *Jeremy played*, *the cat asked* are extracted using *nsubj*, *nsubjpass* and *nn* dependency relationships, where *Jeremy* and *the cat* are the subjects and *played*/*asked* are the verbs associated with them.

Verb-Object (verb-noun) clauses are extracted using the dobj relationship. e.g. For the sentence "The sultan *asked Alladin*" we get the direct object dependency *dobj* (asked, Alladin) where *asked* is the verb and *Alladin* is the noun in the clause. In this case, "*asked Alladin"* is extracted as the verb-object clause.

Consider the example –

```
Then old Mrs. Rabbit took a basket and her umbrella,
to the baker's.
```

The Stanford dependency parser returns the following dependencies for this sentence.

```
advmod(took-5, Then-1)
amod(Rabbit-4, old-2)
compound(Rabbit-4, Mrs.-3)
nsubj(took-5, Rabbit-4)
root(ROOT-0, took-5)
```

```
det(basket-7, a-6)
dobj(took-5, basket-7)
cc(basket-7, and-8)
nmod:poss(umbrella-10, her-9)
conj(basket-7, umbrella-10)
case(baker-14, to-12)
det(baker-14, the-13)
nmod(took-5, baker-14)
case(baker-14, 's-15)
```

The dependencies that the system extracts from this sentence are marked in bold. We observe here that the subject in nsubj is *Rabbit* which is only a part of the character's full name *Mrs. Rabbit*. Thus, in order to extract the complete phrase that is used to identify the character, we check if the dependencies prior or after our four main dependencies (nsubj, nsubjpass, nn, dobj) are *amod, compound* or *det* then we use these three dependencies to retrieve the full name. e.g. The dependency *compound(Rabbit-4, Mrs.-3)* in the example above is used to extract the character name *Mrs. Rabbit.* Similarly, *det(basket-7, a-6)* is also used to retrieve *a basket*.

Our output after the first step for the above example sentence will be two clauses extracted:

- Subject-verb `Mrs. Rabbit took`
- Verb-object `took a basket`

We observe here that the object in the verb-object clause is not a character. In order to filter such subject/objects we find all the verbs from this set of subject-verb and verb-object clauses that are associated with human activity.

Verbs associated with human activity are retrieved in two ways:

4.3.1 Using Derivationally Relational Form

Derivationally Related Form (DRF) is defined in the WordNet glossary as: [7]

> `Terms in different syntactic categories that have the same root form and are semantically related.`

[7] https://wordnet.princeton.edu/documentation/wngloss7wn

We use the derivational related form to check whether a verb has a syntactic category that is related to a noun associated with humans. This relationship is extracted by selecting the verbs that contain the phrases “a person who”, “someone who”, “to whom”, and “one who”.

In our example subject-verb clause *Mrs. Rabbit took* we find the derivational related form of the verb “took”. A subset of the output returned is shown below:

```
[Word: [POS: noun] [Lemma: occupation] [Synset: [Offset: 15166446]
[POS: noun] Words: occupation -- (the period of time during which
a place or position or nation is occupied; "during the German
occupation of Paris")] [Index: 0]]
[Word: [POS: noun] [Lemma: guide] [Synset: [Offset: 10761478]
[POS: noun] Words: usher, guide -- (someone employed to conduct
others)] [Index: 1]]
[Word: [POS: noun] [Lemma: leader] [Synset: [Offset: 9646208]
[POS: noun] Words: leader -- (a person who rules or guides or
inspires others)] [Index: 0]]
[Word: [POS: noun] [Lemma: taking] [Synset: [Offset: 715729] [POS:
noun] Words: pickings, taking -- (the act of someone who picks up
or takes something; "the pickings were easy"; "clothing could be
had for the taking")] [Index: 1]]
```

As we observe, two out of the four DRF descriptions contain the phrases ‘a person who’ and ‘someone who’ that we consider while selecting a verb. In this case, the verb `took` is thus considered as a human activity verb.

4.3.2 Using the Sentence Frame

In the absence of a derivationally related form, the verb is then tested for human activity using the Sentence Frame feature in WordNet.

Each verb synset[8] contains a list of generic sentence frames illustrating the types of simple sentences in which the verbs in the synset can be used.[9] For example, in the sentence “The princed

[8] A set of one or more synonyms that are interchangeable in some context without changing the truth value of the proposition in which they are embedded.
[9] https://wordnet.princeton.edu/documentation/wninput5wn

married Cinderella", the verb-object extracted is "*married Cinderella*" and the sentence frame of the verb *married* returns the following skeletons:

```
Somebody ----s
Somebody ----s somebody
```

For subject-verb phrases we consider a verb to be part of human activity if none of the sentence frame skeletons match the regular expression "Something ----s.*", in other words all of the sentence frames begin with "Somebody ----s" like in the example above. If any of the sentence frames begin with "Something ----" then the verb is not considered as a human activity verb. Similarly, in the case of a verb-object we check if the sentence frame contains ".* something", then the action is not taken by a human and so the verb is eliminated i.e. all the sentence frames must end with "somebody" to be considered as human-activity verbs.

4.3.3 Filtering using Animate being

In order to further fine tune our subject-verb and verb-object clauses, once we have all the verbs that are considered to be human activity verbs, we then add an additional check on their corresponding subject/object (head nouns). We check if the nouns are animate beings. This is done using the hypernym feature in WordNet.

Wordnet defines hypernym as: [10]

> The generic term used to designate a whole class of specific instances.
> **Y** is a hypernym of **X** if **X** is a (kind of) **Y**.

For example, carnivore is a second degree hypernym of cat. The complete example of the entire tree leading up to animate being is shown below.

```
Cat -> feline -> carnivore -> placental -> mammal -> vertebrate -
> chordate -> animate being
```

[10] https://wordnet.princeton.edu/documentation/wngloss7wn

We similarly derive this nested direct hypernym for every noun associated with a human activity verb.

Finally, the noun phrase of the noun that is an ‘animate being’ is then considered as a character name used in the story.

5 Framework

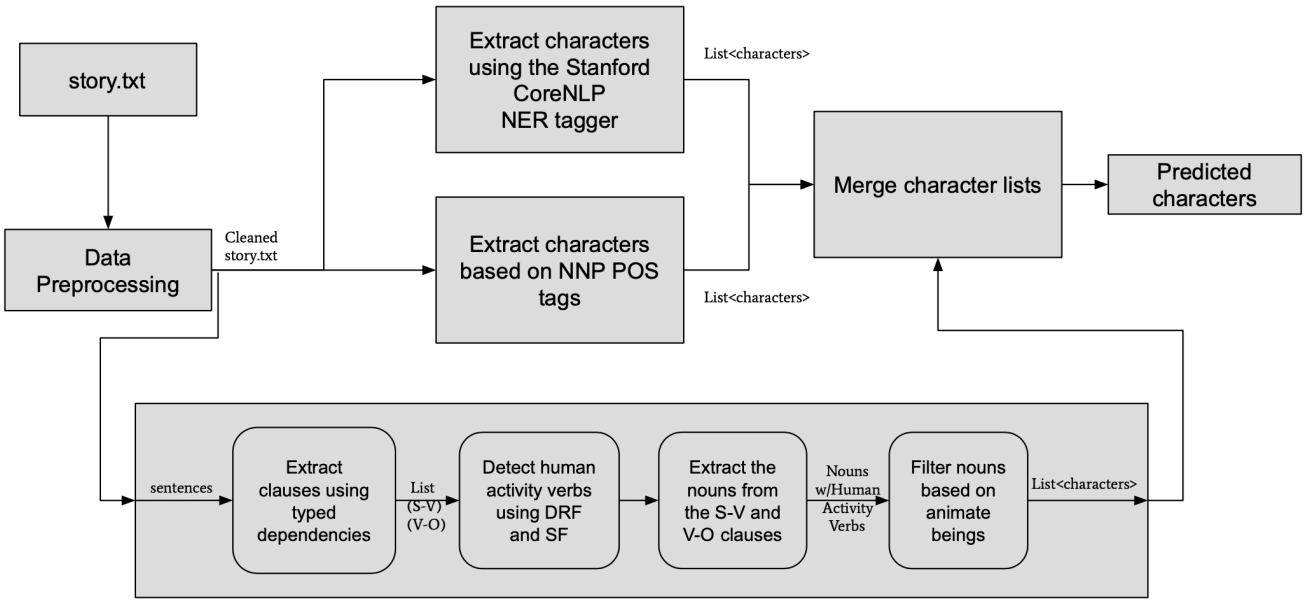

Figure 1 Framework of the system

The complete framework of the system is shown in Figure 1. Each story first passes through a data pre-processing module where it is cleaned up. The extra tags included in the original L^2F corpus text are removed. This cleaned story is then processed in three modules of the corresponding methods.

First, using the Stanford CoreNLP NER tagger, the tags for all the words in the story are retrieved. The PERSON tag is then used to build character names and a list of unique characters is extracted.

Second, using proper noun POS tags (NNP) we find a second list of named characters in the story.

Finally, in the third module, the story is broken into sentences. For each sentence we get its typed dependencies. Nsubj, nn, nsubjpass and dobj dependencies are extracted to build a list of subject-verb and verb-object clauses. The system then finds the derivationally related forms(DRF) and sentence frame(SF) for the verbs in the subject-verb(S-V) and verb-object(V-O) lists using Java WordNet Library.

The result of this is a list of human activity related verbs. The corresponding nouns for these verbs in the S-V and V-O lists are then filtered to find only the nouns that are animate beings. A list of characters is built from the remaining noun phrases.

The list of characters from each of the three modules is then merged to build the final output of predicted characters in the story.

6 Evaluation

The system was evaluated using the evaluation metrics *precision*, *recall*, *f-measure* and *kappa*. Precision is the metric for binary classifier that measures the correctness among all positive labels whereas recall measures how many positive labels are successfully predicted among all positive labels. F-measure is the weighted harmonic mean of precision and recall. We also calculate inter-coder agreement between the gold standard and the system. The system is evaluated in three ways. First, we evaluate the system using noun phrases as the unit of evaluation. We calculate how many of the noun phrases contain a character identified by the system. Second, we consider only the head nouns from the noun phrase for evaluation. Finally, we evaluate the system for each of the six writers and present the results. The McNemar's test is used to calculate significance of the results.

6.1 Baseline

The results of the Stanford CoreNLP Named Entity Recognizer are considered as the baseline to evaluate our system. It was observed that the baseline gives good precision but does not guarantee results across the corpus.

Figure 2 shows the results from the baseline for the development set and Figure 3 shows the results for the test set. We observe that for development set 3 out of 6 stories do not have any characters detected and for test set 2 out of the 6 stories do not have any characters detected by the baseline.

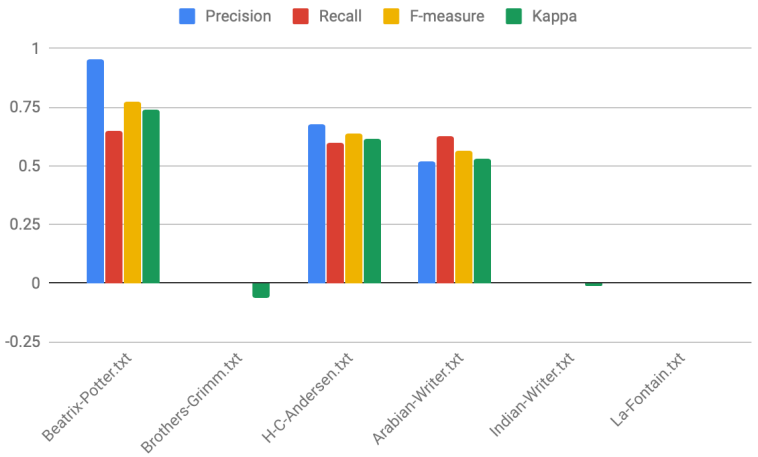

Figure 2 Baseline results for development set

Analyzing these stories, we observed that the results indicate that these stories do not use proper nouns to identify characters but rather make use of noun phrases like *an old woman* which are not detected by the baseline. Hence, we observe f-measure = 0 and negative kappa for such stories.

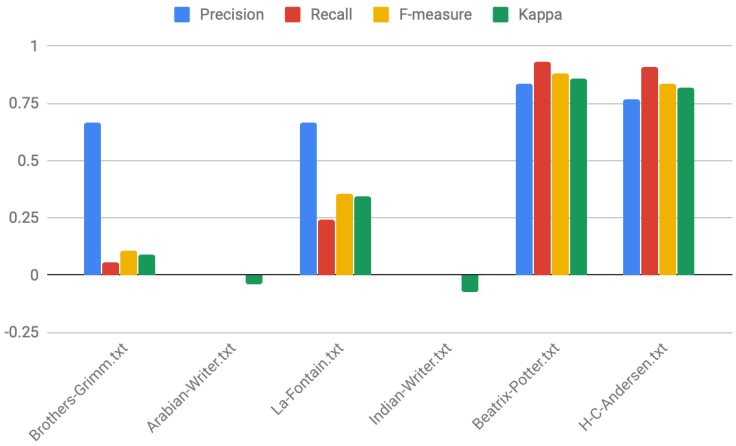

Figure 3 Baseline results for test set

6.2 Evaluation using Noun phrases

The system was evaluated using noun phrases. Noun Phrases (NP) are extracted using the Stanford CoreNLP Parser. For each NP in the story, we check if a character from the gold standard is contained in the noun phrase and if it is also predicted by the system as a character then it is considered as true positive. If the character is present in the gold standard but not predicted by the system, then it is false negative. Similarly, if a character is not present in the gold standard but identified by the system then it is false positive and if neither the system nor the gold standard has any characters contained in the noun phrase then it is true negative.

6.2.1 Overall performance

Figure 4 shows the results for the training set. We observe that f-measure increases by 0.002 but the kappa value decreases by 0.03 decrease. Precision decreases by 0.22 while recall increases by 0.27.

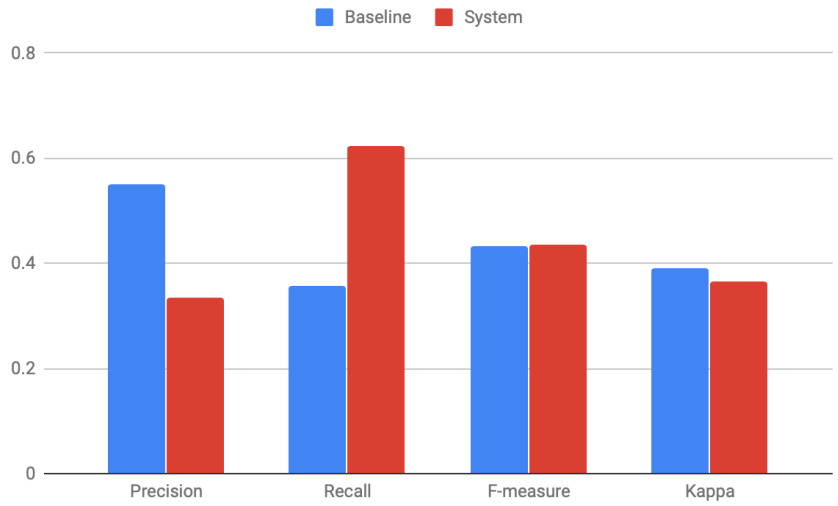

Figure 4 Comparison of results of training set

The decrease in precision is because the system labels a lot of incorrect noun phrases as positives, while the increase in recall indicates that more characters were correctly identified by the system. This shift in identifying more characters could be better for applications as it would be easier for us to manually eliminate nouns that are identified incorrectly, than to go through the story and identify characters that have been missed to be identified by the system.

Moreover, the baseline failed to identify any characters in 7 out of the 18 stories while the proposed system extracted characters for all the 18 stories.

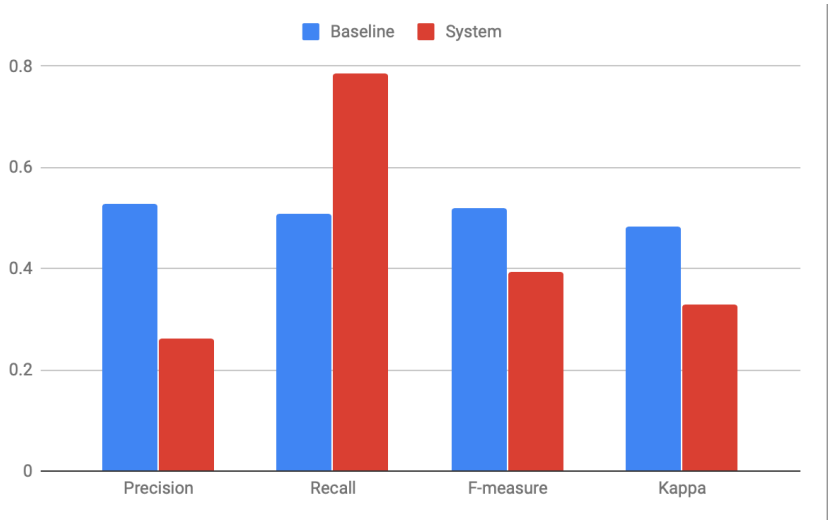

Figure 5 Comparison of results for development set

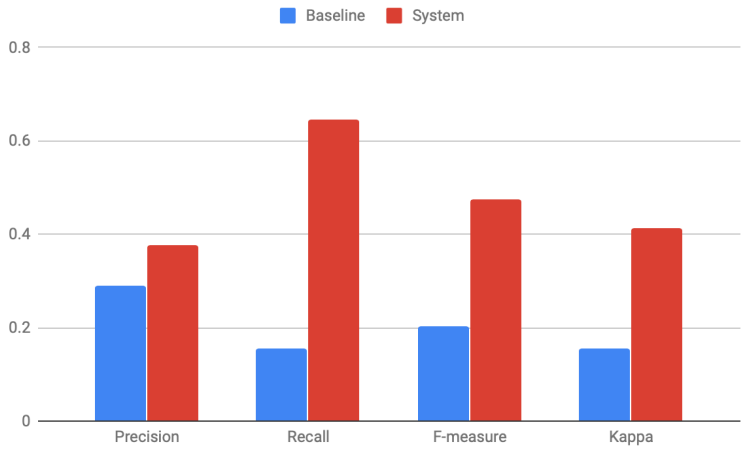

Figure 6 Comparison of results for test set

In case of the development set, we observe that though the recall is high for the system but there is a drop in the values of other measures as shown in Figure 5. However, the test set showed better results for all the measures. Figure 6 shows the comparison of results between baseline and system for the test set.

The system thus seemed to varying performance than the baseline, but it did perform well in terms of reliability. It returned results for all the stories unlike the baseline and performed better than the baseline for the test set.

6.2.2 Evaluating individual features

The human-activity verb method uses three features to detect characters: the derivationally related form, the sentence frame and whether the nouns are animate beings. These features were evaluated to measure the performance of each feature and their combination on the system.

The results are shown in Table 2. We observed that sentence frame showed better performance than derivationally related form when used for detecting characters. When used together, those two features improved the recall and f-measure by 0.01. We also evaluated the performance of the system when derivationally related forms are applied only to subject-verb clauses and sentence frame is applied to verb-objects. We observed that the results were not very different when we apply both sentence frames and derivationally related forms to subject-verb clauses.

Features	Precision	Recall	F-measure	Kappa
DRF *	0.25	0.78	0.38	0.32
DRF (s-v) + SF* (v-o)	0.26	0.79	0.39	0.33
SF	0.27	0.64	0.38	0.32
DRF + SF without animate being	0.21	0.88	0.34	0.28
DRF + SF with animate being	0.26	0.79	0.39	0.33

Table 2 Comparison of results of each feature in human-activity verb method
**DRF-Derivationally related form, SF-Sentence Frame, s-v Subject-verb, v-o Verb-object*

The use of "animate being" showed better performance than not using animate being feature. Though the recall dropped by 0.09, the precision, f-measure and kappa increased by 0.05. We thus retained the use of animate being while detecting characters. Overall the kappa value for all the features was found to be much lower than a human annotators kappa of 0.62.

6.3 Evaluation Using Nouns

The system was also evaluated by measuring the performance on just the nouns. Extracting all the nouns from the story, we check if the system marks that noun as a character or not. For example, if the gold standard and the system predicted "the old king" as character, then only the noun "king" is used from this phrase is used for evaluation. The head noun from the characters were extracted and the results were evaluated based on the total no. of nouns.

Figure 7 shows the result for the training set. The recall, f-measure and kappa show better results for the system than the baseline. This shows that the system performed better when detecting both proper and common nouns that are characters.

In comparison, the development set showed lower performance for the system than the baseline as shown in Figure 8. The decrease in f-measure could be seen as a result of better precision that was observed in the baseline.

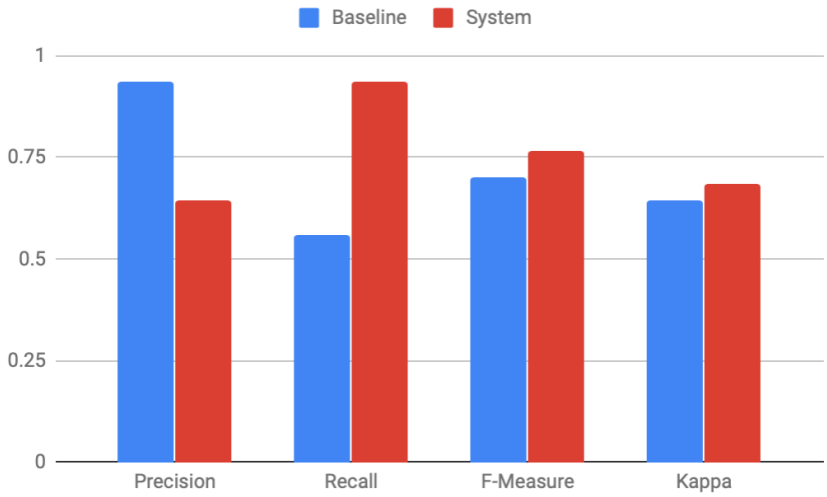

Figure 7 Comparison of results for training set

	Precision	**Recall**	**F-measure**	**Kappa**
Baseline	0.935	0.559	0.7	0.643
System	0.645	0.936	0.767	0.684

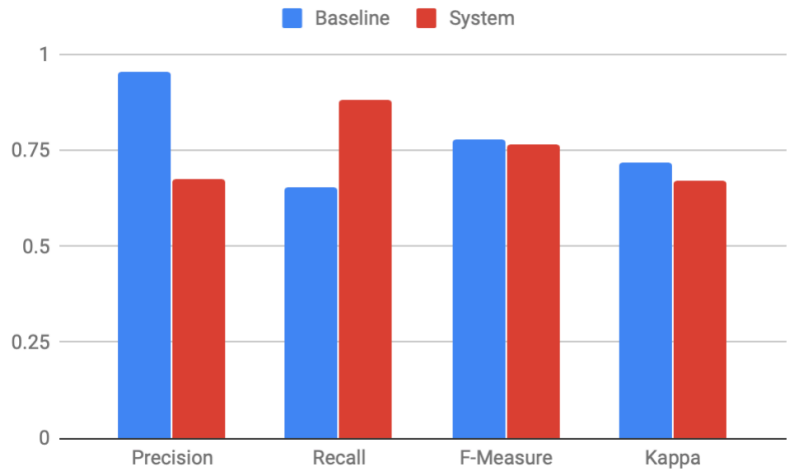

Figure 8 Comparison of results for development set

	Precision	**Recall**	**F-measure**	**Kappa**
Baseline	0.956	0.654	0.777	0.720
System	0.674	0.884	0.765	0.672

A closer evaluation of the development test stories revealed that 4 out of 5 stories for which the baseline returned results had greater than 80% precision, whereas the system's precision dropped to less than 72% across all stories though showing better recall but not good enough to get an overall good f-measure.

Analyzing the stories in the development set we observed that half of the stories used proper nouns heavily to identify characters. This could be the reason why the baseline showed better precision and hence resulting in better f-measure.

Figure 9 shows the results of the test set which indicate significantly better results than baseline. The stories in the test set were analyzed. I found that 5 out of the 6 stories made use of common nouns to identify the characters, which could be why the baseline showed poor recall while the system showed better performance.

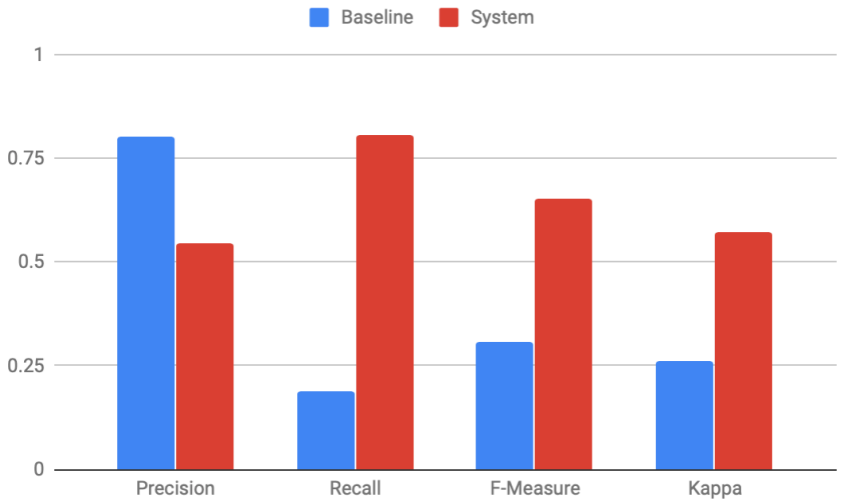

Figure 9 Comparison of results for test set

	Precision	**Recall**	**F-measure**	**Kappa**
Baseline	0.803	0.189	0.306	0.262
System	0.546	0.806	0.651	0.572

In conclusion, the evaluation showed that though the baseline is better for stories where proper nouns are used to identify the characters, the proposed system performed better when common

nouns were used. Overall, especially in the presence of a corpus with mixed stories, the system would be preferred since it was more reliable in detection of characters than the baseline.

6.4 Evaluation based on individual writers

The system was evaluated based on its performance on the corpus of the six writers. The evaluation was performed both using noun phrase and nouns. The results are as shown in Figure 10.

The stories by Beatrix potter, H.C. Anderson, and La Fountain perform better using the baseline than the system. These writers mostly used proper nouns to identify characters. However, the system showed better results across all writers when evaluated using only the head nouns. This shows that the system was able to better detect when a noun is referring to a character irrespective of whether the noun was a proper or common noun.

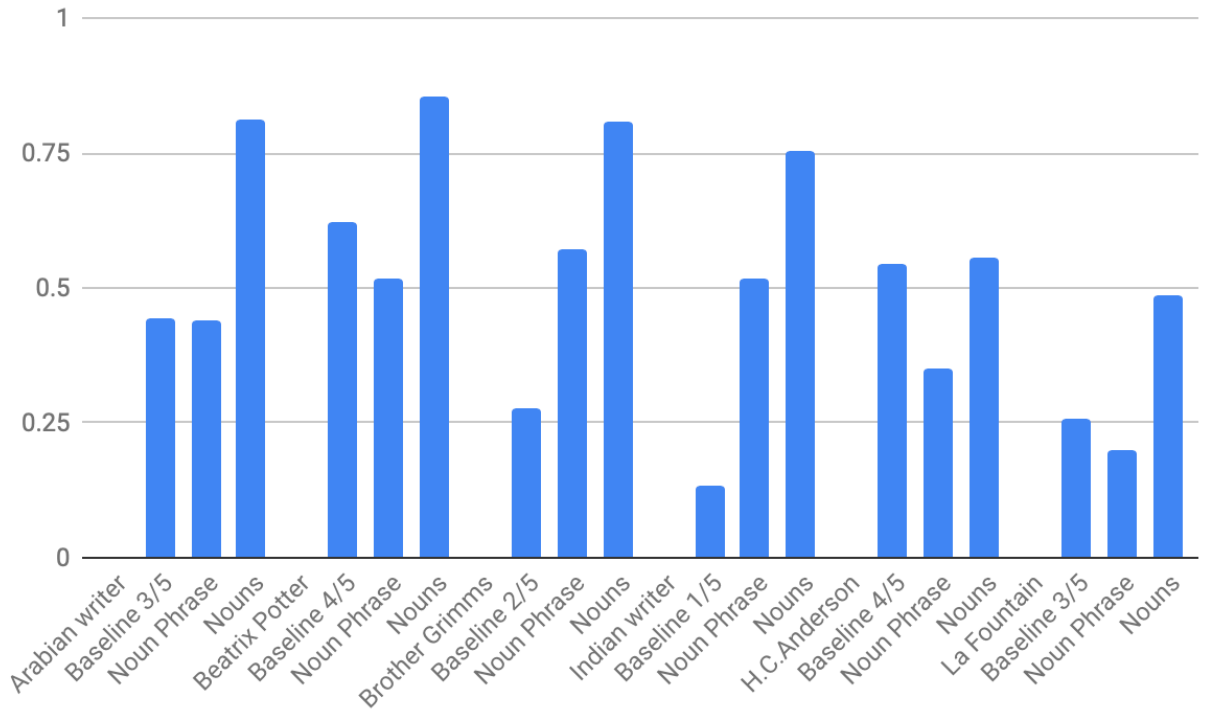

Figure 10 F-measure comparison across all writers

6.5 Statistical Significance using McNemar's test

McNemar's test [8] is a statistical test used on paired nominal data. It is used to compare the accuracy of two models. In our project, we use this to compare the baseline and the implemented system.

The characters identified by human annotators are considered as the gold standard. We calculate how many of these characters are identified correctly by the baseline but not identified by the system. Similarly, we count the number of characters that are identified by the system but not by the baseline. Using the counts, a 2X2 contingency matrix was built as shown in Table 3.

We can see that the system got 282 predictions right that the baseline got wrong. Vice versa, the baseline got 10 prediction right that the system got wrong. Thus, based on this 282:10 ratio, we may conclude that the system performs substantially better than the baseline. [11]

		System		
		Character	Not a character	
Baseline	Character	75	10	85
	Not a character	282	322	604
		357	332	689

Table 3 Contingency table for baseline vs system

The P value based on McNemar's test with continuity correction [9] was calculated. The two tailed p value is less than 0.0001. By conventional criteria, this difference is considered to be extremely statistically significant. [12]

[11] http://rasbt.github.io/mlxtend/user_guide/evaluate/mcnemar/
[12] https://www.graphpad.com/quickcalcs/mcNemar2/

7 Conclusions

The proposed system presents a framework to detect characters in folktales. We use Goh et al.'s proposed method to detect characters in fairytale with a simplified implementation and analyze the system on a broader corpus of folktales. In addition, the system also detects proper nouns that are used to identify characters in fairytales which was one of the drawbacks of Goh et al.'s system.

The evaluation of the system showed that the system is much more reliable when compared to the baseline. We were able to retrieve characters from all the stories in the corpus and also showed that the system performed significantly well especially in the detection of head nouns. We also presented evaluation of the system based on different features that are used to filter subject-verb and verb-object clauses in order to extract the characters in a story. The best combination of these features was identified as a result. McNemar's test with continuity correction was used to show that the results of the system are statistically significant.

However, the proposed system does not identify the character when the story is being told in a first-person narrative. In this case, it cannot reference the "I" that is used. It also does not do well when a single character is being referenced using two completely different names. For example, in the progression of a character from 'the prince' to 'the king', the system identifies these as two different characters.

The system could be useful in applications where character detection is required for a mixed corpus. In case of a corpus which uses mostly proper nouns, the baseline method might be sufficient. But since in a real-world scenario one is unaware of the type of text present in a large corpus, the proposed system could yield better results.

In the future, the system can be further improved by tweaking the human-activity verb method in order to reduce false positives. This could improve the precision of the system and result in better overall performance. We could also improve the co-reference resolution of the system and include detection of characters in first-person narratives, thus, aiming to build a complete system for detection of character in folktales.

8 References

1. Manish Gupta, Piyush Bansal, and Vasudeva Varma, “CharBoxes: A System for Automatic Discovery of Character Infoboxes from Books” *in Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*

2. F. Karsdorp, P. van Kranenburg, T. Meder, and A. van den Bosch, “Casting a spell: Identification and ranking of actors in folktales,” in *Proceedings of the 2nd Workshop on Annotation of Corpora for Research in the Humanities*, Lisbon, Portugal, 2012.

3. D. Suciu and A. Groza, “Interleaving ontology-based reasoning and natural language processing for character identification in folktales,” in *IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP2014)*, Cluj-Napoca, Romania, 2014, pp. 67–74.

4. Hui-Ngo Goh, Lay-Ki Soon, and Su-Cheng Haw, “Automatic identification of protagonist in fairy tales using verb”, in Pang-Ning Tan, Sanjay Chawla, Chin Ho, and James Bailey, editors, Advances in *Knowledge Discovery and Data Mining*, volume 7302 of Lecture Notes in Computer Science, pages 395–406. Springer Berlin / Heidelberg, 2012.

5. Chak Yan Yeung and John Lee, “Identifying Speakers and Listeners of Quoted Speech in Literary Works” in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (pp. 325-329).

6. Paula Vaz Lobo, David Martins de Matos, “Fairy Tale Corpus Organization Using Latent Semantic Mapping and an Item-to-item Top-n Recommendation Algorithm”, In *Language Resources and Evaluation Conference - LREC* 2010, European Language Resources Association (ELRA), Malta, May 2010

7. Princeton University "About WordNet." WordNet. Princeton University. 2010.

8. McNemar, Quinn, 1947. "Note on the sampling error of the difference between correlated proportions or percentages". Psychometrika. 12 (2): 153–157.

9. Edwards AL: Note on the “correction for continuity” in testing the significance of the difference between correlated proportions. Psychometrika. 1948, 13 (3): 185-187.