Information retrieval in folktales using natural language processing

Adrian Groza and Lidia Corde*
*Intelligent Systems Group,
Department of Computer Science,
Technical University of Cluj-Napoca, Romania
Adrian.Groza@cs.utcluj.ro,Lidia.Corde@cs-gw.utcluj.ro

***Abstract*—Our aim is to extract information about literary characters in unstructured texts. We employ natural language processing and reasoning on domain ontologies. The first task is to identify the main characters and the parts of the story where these characters are described or act. We illustrate the system in a scenario in the folktale domain. The system relies on a folktale ontology that we have developed based on Propp's model for folktales morphology.**

***Index Terms*—Natural language processing, ontologies, literary character, folktales.**

I. Introduction

Recognising literary characters in various narrative texts is challenging both from the literary and technical perspective. From the literary viewpoint, the meaning of the term "character" leaves space to various interpretations. From the technical perspective, literary texts contain a lot of data about emotions, social life or inner life of the characters, while they are very thin on technical, straight-forward messages. To infer the character type from literary texts might pose problems even to the human readers [4].

Interactions between literary characters contain rich social networks. Extracting these social networks from narrative text has gained much attention [13] in different domains such as literary fiction [6], screenplays [1], or novels [9], [2].

Our aim is to correctly determine the relationships of a character in a tale and to find its role upon the development of the story. In line with [16], the first task is to identify the parts of the story where that character is involved. Our approach relies on interleaving natural language processing and ontology-based reasoning. We enact our method in the folktale domain.

Information extraction systems usually have three components responsible for: named entity recognition, co-reference resolution and relationship extraction. These modules are integrated in a pipeline, in a layered manner, given that each task will use information provided by the previous neighbor. Natural language processing has been applied in the domain of folktales [14], [8]. Formal models for folktales have been proposed in [12], [15]. Character identification in folktales have been approached in [17], [19].

The remaining of the paper is organized as follows: Section II presents the ontology that we developed for modeling the domain of folktales. Section III depicts the architecture of our system. Section IV illustrates our method to extract knowledge about characters. Section V presents the experimental results on seven folktales. Section VI browses related work, while section VII concludes the paper.

978-1-4673-8200-7/15 $31.00 © 2015 IEEE

TABLE I
Main characters in the Propp's model.

Name	*Description*
Villain	The opponent of the hero - often the representation of evil.
Dispatcher	The person that sends the hero into the journey, or the person that informs the hero about the villainy.
(Magical) Helper	The one that helps the hero into its journey.
Princess or Prize	It represents what the hero receives when it is victorious.
Donor	Prepares the hero for the battle.
Hero	The main character in a story - often the representation of good.
False hero	The one that tries to steal the prize from the hero, or tries to marry the princess.

II. Engineering the folktale ontology

To support reasoning in the folktale domain, we developed an ontology used to extract knowledge regarding characters. We assume the reader is familiarised with the syntax of Description Logic (DL). For a detailed explanation about families of description logics, the reader is referred to [3].

To support character identification and reasoning on these characters we need structured domain knowledge. Hence, we developed an ontology for the folktale domain as shown in Fig. 3. Our folktale ontology formalizes knowledge from three sources: 1) the folktale morphology as described by the Propp model [15]; 2) various entities specific to folktales (i.e., animals, witch, dragons); and 3) common family relations (i.e., child, fiancee, groom). In the following, these three knowledge sources are detailed:

a) Folktale morphology: Firstly, we rely on the Propp's model [15] of the folktale domain. In the Propp's model the story broke down into several sections. Propp demonstrated that the sequence of sections appears in the same chronological order in Russian folktales. Propp identified a set of character types that appear in most of the folktales (see Table I).

The corresponding formalization in Description Logic appears in Fig. 1, where the characters are divided in nine types

A_1 Agent $\sqcup$ Donor $\sqcup$ FalseHero $\sqcup$ Hero $\sqcup$ Prisoner $\sqcup$ Villain $\sqcup$ Dispatcher $\sqcup$ MagicalHelper $\sqcup$ Princess $\sqsubseteq$ Character
A_2 Hero $\sqcap$ Villain $\sqsubseteq$ FalseHero
A_3 PositiveCharacter $\sqcup$ NegativeCharacter $\sqsubseteq$ Character
A_4 Villain $\sqcup$ FalseHero $\sqcup$ Prisoner $\sqsubseteq$ NegativeCharacter
A_5 Hero $\sqcup$ MagicalHelper $\sqcup$ Agent $\sqcup$ Donor $\sqcup$ Prisoner $\sqcup$ Dispatcher $\sqsubseteq$ PositiveCharacter

Fig. 1. Formalising the Propp's model of folktales.

A_{21} Bear $\sqcup$ Bird $\sqcup$ Dog $\sqcup$ Duck $\sqcup$ Frog $\sqcup$ Horse $\sqcup$ Lion $\sqsubseteq$ SingleAnimal
A_{22} Enchantress $\equiv$ Witch
A_{23} Enchantress $\sqsubseteq$ Woman $\sqcap$ SingleSocialStatus
A_{24} Giant $\sqsubseteq$ Supernatural
A_{25} Goldsmith $\sqcup$ Helmsman $\sqsubseteq$ SingleSocialStatus
A_{26} King $\sqsubseteq$ SingleSocialStatus
A_{27} Oven $\sqsubseteq$ Object
A_{28} Prince $\equiv$ Son $\sqcap$ $\exists$hasParent.King $\sqcup$ $\exists$hasParent.Queen
A_{29} Prince $\sqsubseteq$ SingleSocialStatus
A_{30} Princess $\equiv$ Daughter $\sqcap$ $\exists$hasParent.King $\sqcup$ $\exists$hasParent.Queen

Fig. 2. Common entities in the folktale domain.

(axiom 1). In axiom 2, a false hero is a hero who is also a villain. Axiom 3 divides the characters into negative and positive ones. Note that positive and negative characters are not disjoint, as for instance the concept *Prisoner* belongs to both sets.

b) Folktale main entities: Secondly, the common entities appearing in folktales were formalized in Fig. 2. The axioms depict the animals (axiom 21), witches or enchantresses which are women with a single social status (axioms 22 and 23), and supernatural characters like *Giant* in axiom 24. Specific characters like *Goldsmith* or *King*, and various objects (i.e. *oven*) are also modeled. A prince is defined in axiom 28 as a son that have a parent either a king or a queen. Similarly, the princess is a daughter with at least on parent of type king or queen (axiom 30).

c) Family relationships in folktale: Fig. 4 lists part of the family relationships adapted to reason in the folktale domain. A significant part of these relationships are correlated with the recurrent theme of the main character who is finding his bride or fiancee.

To facilitate reasoning on the ontology, we allow several extensions of the $\mathcal{ALC}$ version of description logics [3]. Using role inheritance we can specify that the role *hasFather* is more specific than the role *hasParent*. Hence, if we find in the folktale that a character has a father, the system deduces based on role inheritance that the character has also a parent. Similarly, inverse roles like *hasChild* and *hasParent* are used to infer new knowledge based on the partial knowledge extracted by natural language processing. If we identify that two individuals are related by the role *hasChild*, the system deduces that those individuals are also related by the role *hasParent*. The domain restriction specifies that only persons can have brothers. The range restriction constraints the range of the role *hasGender* to the concept *Gender*.

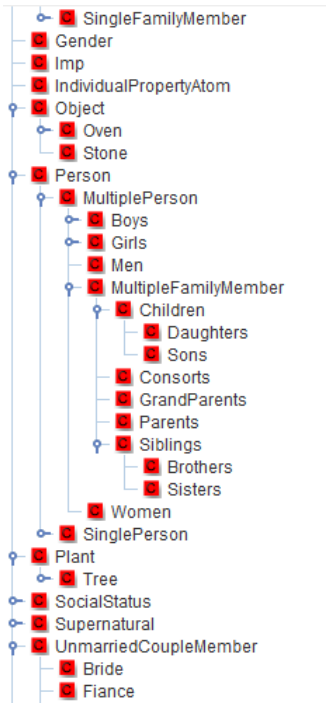

Fig. 3. Folktale ontology.

A_{50} Boy $\sqsubseteq$ SinglePerson
A_{51} Boys $\sqsubseteq$ MultiplePerson
A_{52} Bride $\equiv$ Fiancee
A_{53} Bride $\sqsubseteq$ UnmarriedCoupleMember
A_{54} Brother $\sqsubseteq$ Sibling $\sqcap$ Male
A_{55} Daughter $\equiv$ Girl $\sqcap$ $\sqcap$Child $\exists$hasParent.Parent
A_{56} Father $\equiv$ Man $\sqcap$ $\exists$hasChild.Child
A_{57} Fiance $\equiv$ Groom
A_{58} Fiance $\sqsubseteq$ UnmarriedCoupleMember
A_{59} Fiancee $\equiv$ Bride
A_{60} Fiancee $\sqsubseteq$ UnmarriedCoupleMember
A_{61} Girl $\equiv$ Maiden
A_{62} Girl $\sqsubseteq$ SinglePerson
A_{63} Husband $\sqsubseteq$ Consort

Fig. 4. Family relationships in the folktale domain.

TABLE II
EXPLOITING ROLE CONSTRAINTS TO REASON ON THE ONTOLOGY.

Extensios of ALC	*Folktale examples*
Role inheritance	hasBrother $\sqsubseteq$ hasSibling, hasFather $\sqsubseteq$ hasParent, hasHusband $\sqsubseteq$ hasConsort
Inverse roles	hasHusband $\equiv$ hasWife$^-$, hasChild $\equiv$ hasParent$^-$
Transitive roles	hasSibling^t
Domain restriction	$\exists$hasBrother.$\top$ $\sqsubseteq$ Person
Range restriction	$\top$ $\sqsubseteq$ $\forall$hasBrother.Person, $\top$ $\sqsubseteq$ $\forall$hasGender.Gender
symmetric roles	hasConsort $\equiv$ hasConsort$^-$
cardinality constraints	$\top$ $\sqsubseteq$ $\leq$1 hasGender.Thing

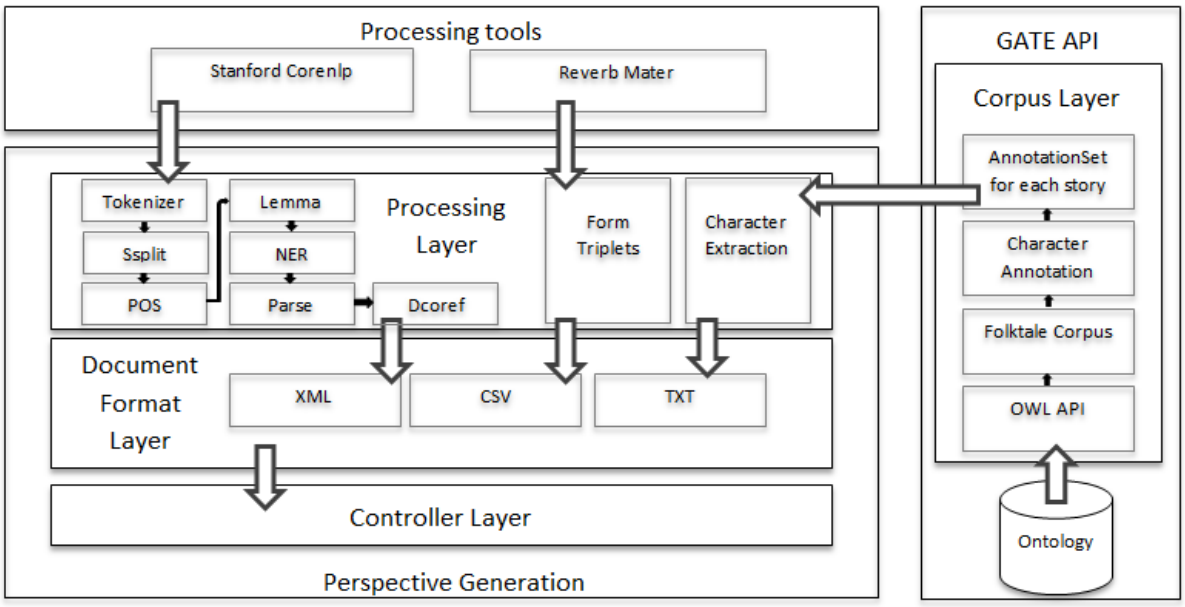

Fig. 5. The System Architecture

III. System Architecture

Extracting knowledge about characters is obtained by interleaving natural language processing (NLP) and reasoning on ontologies. The NLP component is based on GATE text engineering tool [5], while reasoning in DL on the OWLAPI [10], as depicted by the architecture in Fig. 5.

Firstly, the folktale ontology is processed using OWLAPI to generate classes of characters from the ontology into GATE. The folktale corpus is analysed aiming to populate the ontology and to annotate each folktale with the identified named entities. In parallel to the annotation process, the Stanford parser creates the coreference information files. The task is challenging, as even a human might have a problem in decoreferencing some of the sentences, as example 1 illustrates.

Example 1. *"The Smiths went to visit the Robertsons. After that, they stayed home, watching tv.", where "they" might be tied to the Smiths, or the Robertsons, or to both of the families.*

For de-coreferencing, the following pipeline was designed (left part of Fig. 5). The tokenizer groups all the letters into words. Next, the sentence splitter (Ssplit) groups the sequence of tokens obtained in the previous step into sentences. The part of speech (POS) annotation labels all the tokens from a sentence with their POS tags. Lemma annotation generates the word lemmas for all the tokens in the corpus. The next step is to apply named entity recognition (NER) so that the numerical and temporal entities are recognized. This is done using a conditional random fields (CRF) sequence taggers trained on various corpora. The parse function provides a full syntactic analysis for each sentence in the corpora. Finally, the coreference chain annotation (Dcoref) obtains both the pronominal and nominal coreference resolution. After coreference resolution, the stories are updated with the coreference information.

The Reverb information extraction tool [7] is used to generate triplets containing the following structure: ⟨nominal phrase, verb phrase, nominal phrase⟩. For the sentence *"Good heavens, said the girl, no strawberries grow in winter"*, the output of Reverb is exemplified in Table III. In order to obtain the triplets, each sentence has to be POS-tagged and NP-chunked.

IV. Interleaving Natural Language Processing with Reasoning on Ontologies

This section details three algorithms used to identify knowledge about characters. Algorithm 1 identifies characters in the folktale. Algorithm 3 is used for anaphora resolution of the named entities recognized as characters. Algorithm 2 extracts knowledge about characters from the de-coreferences. The execution flow of this pipeline, is presented in Fig. 6.

Natural language processing is enacted to populate the folktale ontology. The extraction Algorithm 1 is performed repetitively on a document, each time using the newly populated ontology file. In this way, the algorithm interleaves reasoning on ontology with natural language processing based on Japes rules [18]. The first step is to apply the Jape rules JN on the folktale corpus aiming to identify all the definite and indefinite nominal phrases. Given that the characters are nominal phrases, this first step returns all the information needed, plus some extra phrases that have to be filtered out.

Next, the Jape rules JC are enacted to select candidate characters from the set of nominal phrases previously identified. For each character found, a set of rules JR is used to match the character against a concept in the ontology.

TABLE III
EXTRACTING TRIPLETS FROM FOLKTALES USING REVERB.

Original Sentence	*Nominal Phrase (arg1)*	*Verb Phrase (arg2)*	*Nominal Phrase (arg3)*	*Extraction Confidence*	*POS tags*	*Chunk tags*
10	*3*	*4*	*5*	*9*	*11*	*12*
Good heavens, said the girl, no strawberries grow in winter.	no strawberries	grow in	winter	0.505	JJ NNS , VBD DT NN , DT NNS VB IN NN .	B-NP I-NP O B-VP B-NP I-NP O B-NP I-NP
The king's daughter began to cry , for daughter was afraid of the cold frog which daughter did not like to touch, and which was now to sleep in daughter pretty, clean little bed.	daughter	was afraid of	the cold frog	0.691	DT NN POS NN VBD TO VB , IN NN VBD JJ IN DT JJ NN WDT NN VBD RB IN TO VB , CC WDT VBD RB TO VB RP NN RB , JJ JJ NN .	B-NP I-NP I-NP I-NP B-VP I-VP I-VP O B-PP B-NP B-VP B-ADJP B-PP B-NP I-NP I-NP B-NP I-NP B-VP O O B-VP I-VP O O B-NP B-VP B-ADVP B-VP I-VP B-NP I-NP B-ADVP O B-NP I-NP I-NP O
When everything was stowed on board a ship, faithful John put on the dress of a merchant, and the king was forced to do the same in order to make king quite unrecognizable.	John	put on	the dress of a merchant	0.876	WRB NN VBD VBN IN NN DT NN , NN NNP VBD IN DT NN IN DT NN , CC DT NN VBD VBN TO VB DT JJ IN NN TO VB NN RB JJ .	B-ADVP B-NP B-VP I-VP B-PP B-NP B-NP I-NP O B-NP B-NP B-VP B-PP B-NP I-NP I-NP I-NP I-NP O O B-NP I-NP B-VP I-VP I-VP I-VP B-NP I-NP B-SBAR O B-VP I-VP B-NP B-ADJP I-ADJP O
Sons each kept watch in turn, and sat on the highest oak and looked towards the tower.	each	kept watch in	turn	0.880	NNPS DT VBD NN IN NN , CC VBD IN DT JJS NN CC VBD IN DT NN .	O B-NP B-VP B-NP B-PP B-NP O O B-VP B-PP B-NP I-NP I-NP O B-VP B-PP B-NP I-NP O
Rapunzel grew into the most beautiful child under the sun.	Rapunzel	grew into	the most beautiful child	0.830	NNP VBD IN DT RBS JJ NN IN DT NN .	B-NP B-VP B-PP B-NP I-NP I-NP I-NP B-PP B-NP I-NP O
The king's son ascended, but instead of finding son dearest rapunzel, son found the enchantress, who gazed at son with wicked and venomous looks.	the enchantress	gazed at	son	0.586	DT NN POS NN VBD , CC RB IN VBG NN NN NN , NN VBD DT NN , WP VBD IN NN IN JJ CC JJ NNS .	B-NP I-NP I-NP I-NP B-VP O O B-PP I-PP B-VP B-NP I-NP I-NP O B-NP B-VP B-NP I-NP O B-NP B-VP B-PP B-NP B-PP B-NP I-NP I-NP I-NP O

Input : O_f - Folktale ontology;
S - Corpus of folktales;
JN - Jape rules to identify definite and indefinite nominal phrases;
JC - Jape rules to identify candidate characters;
JR - Jape rules to identify character's relation to the ontology;

Result: C: Set of annotated characters;

$C \leftarrow \emptyset$;
$NP \leftarrow$ applyRules(JN, S);
while *applyRules(*JC, S, NP*)* $\neq$ *null* **do**
  $NC \leftarrow$ applyRules(JC, S, NP);
  $Rel \leftarrow$ applyRules(JR, S, NC);
  foreach $r \in Rel$ **do**
    foreach *concept from* r **do**
      if *checkCast(*NC, *concept)* **then**
        cast(NC, *concept*);
      end
    end
  end
  while *is_referred(*S, NC*)* **do**
    Ref = getReference();
    link(NC, Ref);
  end
  $C \leftarrow C \cup NC$;
end

Algorithm 1: Character extraction algorithm.

After identifying a concept for which the character is an instance, the algorithm exploits reasoning on ontology to identify all atomic concepts to which the character belongs. For instance, a character identified as $Daughter$ will be an instance of $Girl$, $Child$, $Maiden$, $SinglePerson$ (recall Fig. 4). For each concept to which the character belongs, the algorithm looks again in the corpus to see if there are other mentions of the newly introduced character. If this is the case, the character is related with the new knowledge.

Input : S: Corpus of folktales; P: Pipeline configuration for decoreferencing;
FN: List with filenames for each S;
SC: Stanford-CoreNLP command;

Result: D: Decoreferenced texts of files from FN;

$Files$ = run(SC, P, FN);
foreach $file$ *in* $Files$ **do**
  $D \leftarrow S$;
  foreach $coref_group \in file$ **do**
    $rep \leftarrow$ findRepresentative($coref_group$);
    foreach $coref_word \in coref_group$ **do**
      replace(D, $coref_word$, rep);
    end
  end
end

Algorithm 2: Decoreference algorithm.

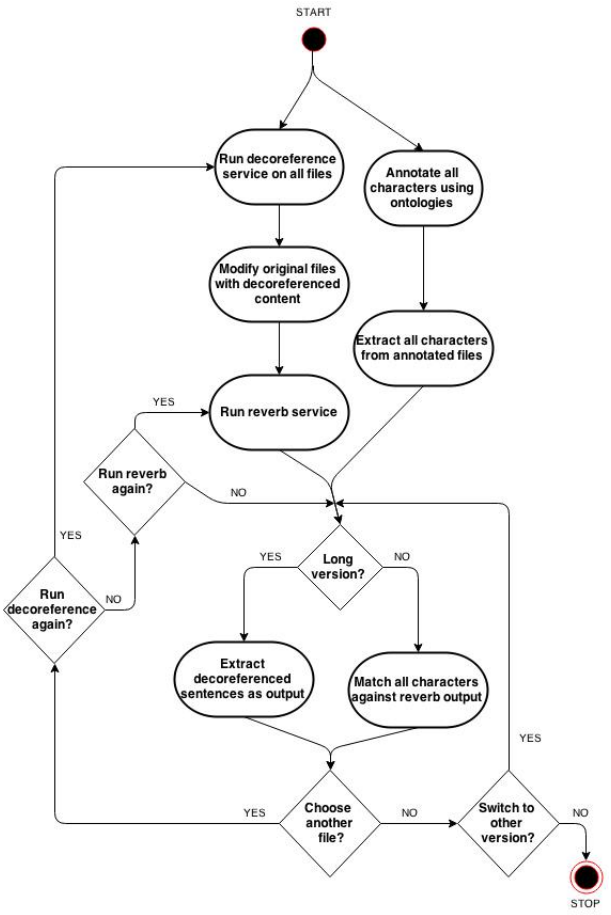

Fig. 6. Main execution phases.

The decoreferencing algorithm (Alg. 2) uses as input the processing pipeline and the folktale corpus. The basic processing steps needed are the following: *tokenize, ssplit, pos, lemma, ner, parse, dcoref*. The decoreferencing algorithm is run on all stories at once, but it generates different output file for each story represented by the filename. In the first step, the Stanford parser applies the execution pipeline on the corpora of folktales. For each resulted file, the algorithm searches for coreference groups. In order to be able to return the modified text, the original text has to be stored in the returning argument of the algorithm. For each coreference group found, firstly the referenced word has to be processed and kept into a variable and then, each coreferenced word found, belonging to the group, has to be replaced in the original text with the referenced variable. In the end, the decoreferenced text for each corpus file is obtained.

Algorithm 3 takes as input the result of algorithms 1 and (alg 2. The set of characters is used as the input, while the decoreferenced texts are used as an environment from which the algorithm extracts the perspective. For each character in the set of characters resulted from the extraction algorithm (alg 1), each line that resulted from reverb execution is processed. From each line, the sentence is extracted based on the output

Input : R: Reverb command;
V: The version indicator. True if long version, false otherwise;
C: Set of characters resulted from algorithm 1;
D: Decoreferenced text resulted from algorithm 2;
Result: P: String containing character's perspective in S;

RR = run(R, D);
if V = *true* **then**
  foreach $c \in C$ **do**
    foreach $line \in RR$ **do**
      $sentence \leftarrow$ getSentence($line$);
      if $c \in sentence$ **then**
        $P \leftarrow P \cup sentence$;
      end
    end
  end
else
  foreach $c \in C$ **do**
    foreach $line \in RR$ **do**
      $triplet \leftarrow$ getTriplet($line$);
      if $c \in triplet$ **then**
        $P \leftarrow P \cup triplet$;
      end
    end
  end
end

Algorithm 3: Finding character's perspective.

format of the Reverb service presented in Table III. If the character, from the character set, is mentioned in the sentence, then the sentence is appended to the output variable. These columns are combined in a triplet, and it is checked to see whether the current character appears is present in this triplet. In this case, the triplet is appended to the output variable. This algorithms score is represented by a subunitary number that represents the confidence that the extraction was correct.

V. Experimental results

A. Running scenario

The system was tested against seven stories (Table V). This section illustrates the results of this pipeline for the secondary character Henry from the story "The frog king". The fragment on which the algorithms were applied is listed in Example 2.

Example 2. *"Then they went to sleep, and next morning when the sun awoke them, a carriage came driving up with eight white horses, which had white ostrich feathers on their heads, and were harnessed with golden chains, and behind stood the young king's servant Faithful Henry. Faithful Henry had been so unhappy when his master was changed into a frog, that he had caused three iron bands to be laid round his heart, lest it should burst with grief and sadness. The carriage was to conduct the young king into his kingdom. Faithful Henry helped them both in, and placed himself behind again, and was full of joy because of this deliverance. And when they had driven a part of the way the king's son heard a cracking behind him as if something had broken. So he turned round*

TABLE IV
ELICITING KNOWLEDGE ABOUT HENRY.

	Character: Henry
1	Henry master was changed into a frog
2	Henry had caused three iron bands
3	faithful Henry helped bands
4	bands placed Henry
5	Henry was full of joy
6	the bands were springing from the heart of faithful Henry

and cried, "Henry, the carriage is breaking." "No, master, it is not the carriage. It is a band from my heart, which was put there in my great pain when you were a frog and imprisoned in the well." Again and once again while they were on their way something cracked, and each time the king's son thought the carriage was breaking, but it was only the bands which were springing from the heart of Faithful Henry because his master was set free and was happy."

The method has two kind of results - one for the long version, and one for the short version. Firstly, the results for the short version are listed in Table IV. Note that the output text is the decoreferenced one - this is the reason why the character might talk about itself in third person. Because of the de-coreferenced version of the stories part of text might not be correct from the human reader perspective. But it is the easiest way to understand the context of a character. Otherwise, it would be hard to see that when the text says "his master", that "his" refers to Henry, as Example 3 bears out.

Example 3. *1. Then companion went to sleep, and next morning when the sun awoke companion, a band came driving up with eight white horses, which had white ostrich feathers on companion heads, and were harnessed with golden chains, and behind stood the young king's servant faithful Henry.*

2. Faithful Henry had been so unhappy when henry master was changed into a frog, that Henry had caused three iron bands to be laid round henry heart, lest heart should burst with grief and sadness.

3. Faithful Henry helped bands both in, and placed Henry behind again, and was full of joy because of this deliverance.

4. Again and once again while you were on you way something cracked, and each time the king's son thought the band was breaking , but it was only the bands which were springing from the heart of faithful Henry because Henry master was set free and was happy."

There are some cases in which there will be no result for a character (Example 4). Given that the character was extracted from the original file, by using Algorithm 1, there is a certainty that the character exists in the story.

Example 4. *When trying to search for the perspective of character "waiting-maid" in the story "Faithful John", the application will not be able to find any solution. In the unmodified text, the son character is introduced in the following way: "She took him by the hand and led him upstairs, for she was the waiting-maid."*

TABLE V
ACCURACY OF THE ALGORITHMS.

Story	*Accuracy*
The Magic Swan-Geese	75%
The Frog King	62%
The King's Son who Feared Nothing	76%
Faithful John	63%
The Twelve Brothers	65%
Rapunzel	74%
The Three Little Men in the Woods	73%
Average	70%

This happens because, when the anaphoric decoreference is run (Algorithm 2), the file is changed in the following way: *"Girl took oh by the hand and led oh upstairs , for girl was the girl ."*. The change happened because the decoreferencing tool interpreted *"the waiting-maid"* as being tied up to the word *"she"*, and, which is tied to *"the girl"* from the following phrase *"Then said the girl ' the princess must see these , girl has such great pleasure in golden things , that girl will buy all you have . '"*. In this way, this character's part will be attributed to the *"girl"*, which is the main character of the story. This situation in which the story is talking about a general character, but only after the main events, the character is finally revealed, is called *cataphora* [11].

B. Accuracy of the method

The accuracy of our method is influenced by: 1) accuracy of character identification; 2) accuracy of identifying co-references; 3) accuracy of Reverb when extracting triplets (the confidence indicator). Each of this services has an accuracy error that will be propagated from one component to another. We performed various tests on the corpus used for character identification, and we obtained an average accuracy of 70% (Table V). When calculating the accuracy, 20 characters were taken into consideration, meaning that for each story, about 3 characters were chosen. These characters were manually selected from the set of characters output by the character extraction system presented in [17], [19]. The characters were selected by choosing 2 main characters and a secondary character for each story.

The testing was performed on seven different stories, and for each story, a set of main characters was chosen. The obtained overall accuracy is 74%, having an overall precision of 90% and a recall of 60%. The results are presented in Fig. 7. Figure 8 depicts the distribution of precision, recall and accuracy over the stories. The values were calculated using the following formulas:

$$precision = \frac{tp}{tp+fp}$$
$$recall = \frac{tp}{tp+fn}$$
$$accuracy = \frac{tp+tn}{tp+fp+tn+fn}$$

where tp means *true positive*, and represents the number of sentences that are found both in the manually annotated set and the test set, tn means *true negative* and represents the number of sentences that are neither in the manually annotated

Story:	The Magic Swan-Geese	The Frog King	The King's Son who feared nothing	Faithfull John	The twelve Brothers	Rapunzel	The three little Men in the Woods
p/story	1	0,95	0,93	0,85	0,80	0,91	0,86
r/story	0,48	0,37	0,81	0,34	0,84	0,54	0,82
a/story	0,74	0,47	0,91	0,60	0,82	0,77	0,86
total prec	0,904						
total rec	0,605						
total acc	0,741						

Fig. 7. Precision, recall, and accuracy for the seven folktales analyzed.

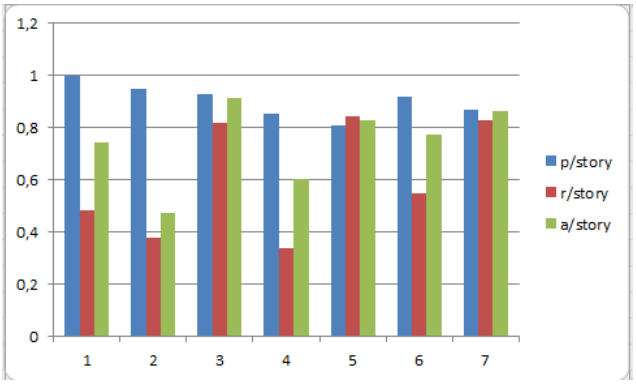

Fig. 8. Comparing precision, recall and accuracy for each story.

set, nor in the test set, fp means *false positive* and represents the number of sentences that are in the test set and not in the manually annotated set, and fn means *false negative* and represents the number of sentences that are in the manually annotated set, but not in the test set. In the folktale context, the tp represents the number of sentences that belong to the character's perspective, all those sentences that involve the character in any way.

The average F-score for the Stanford-CoreNLP of 59.5 influences greatly the performance of the algorithm, as the characters perspective cannot be extracted, given that the character is not seen as being part of the sentence. The accuracy can be improved if a better decoreferencing tool will be used. Other coreference tools are For the anaphoric decoreference, there are several other tools (BART, JAVARAP, GuiTar and ARKref), but, from all, the Stanford-CoreNLP has be highest accuracy percentage.

There is ongoing research in the coreference resolution domain, When calculating the performance scores, the extraction of the correct sentence was considered, and not on the correctness of the extracted sentence. Even though the right sentence was extracted, the information in the sentence will be according to the coreference resolution result. Hence, an error might be observed when reviewing the structure of the sentences. The algorithms performance is also influenced by the scores obtained by the Reverb tool. Also, the named entity recognition has an average precision of 79% and a recall of 72%. These scores do not influence directly the algorithms performance, but they have an effect on the number of characters for which the algorithm will try to find the roles they have on the development of the story. Together, all these scores combined, give the performance scores of the characters perspective in texts.

The current version does not extract information about the characters' roles. The information extracted consists of the character identification, that is presented in [17], [19], and the story involving the character. The story can be presented in a standardized version.

VI. Discussion

We can enact our solution in other domains instead of folktales. We exemplify he following three domains: a) software requirements, b) marketing and c) medical domain.

Consider the domain of *software requirements*, where these requirements are written in natural language. Our system will support the identification of various actors appearing in the requirements document. First, one needs to replace the folktale ontology with a requirement ontology that provides knowledge on use cases, actors, their roles, etc. The same pipeline will be used to: 1) identify main actors (admin, various users, etc) and 2) extract knowledge about various actions these actors are supposed to perform.

Another domain that could benefit from the same pipeline of execution, would be the *marketing domain*. Consider a dataset of product reviews or accommodation places in the tourism domain [20]. The system would extract only the sentences that reference the mentioned item. By having access to all the sentences of interest, further analysis is facilitated without having to process the entire text.

Similar extraction systems have been proposed for the *medical domain* to extract information from clinical narratives. In this line, the MedEx system [21] aims to extract the medication information from clinical narratives. Similarly, there is also the OpenClinical system for assisting health care providers.

In our approach, the extraction algorithm part is separated from the perspective searching part. Therefore, any ontology and any document can be used in order to find the character's or object's perspective in the document.

We tested our method only on seven stories. With a complexity of $O(n^3)$ in sentence length of syntactic parsing, our syntactic based on Stanford parser might be too slow for large corpus as the one of 15099 narratives analysed in [4].

VII. Conclusions

Our method is able to extract knowledge on various characters. Our current accuracy for information extraction in the folktale domain is 74%. The experimental results were obtained for seven stories in the folktale domain. The precision score is above 90%, With an overall recall of only 60%, there are high chances that not all the information regarding a product was extracted.

The developed algorithms aggregate three different services: Firstly, the named entity recognition was implemented by

using an ontology based on Propps formal model. Based of this ontology, and some implemented Jape rules, the characters are extracted from a given story. Secondly, a coreference resolution tool was implemented by enacting anaphoric resolution to eliminate co-referenced words and to replace them with their representative, Thirdly, finding relationships between characters was integrated in order to link two noun phrases with a verbal phrase.

ACKNOWLEDGMENTS

We thank the reviewers for their valuable comments. Part of this work was supported by the Department of Computer Science of Technical University of Cluj-Napoca, Romania.

REFERENCES

[1] A. Agarwal, S. Balasubramanian, J. Zheng, and S. Dash, "Parsing screenplays for extracting social networks from movies," *EACL 2014*, pp. 50–58, 2014.

[2] A. Agarwal, A. Corvalan, J. Jensen, and O. Rambow, "Social network analysis of Alice in Wonderland," in *Workshop on Computational Linguistics for Literature*, 2012, pp. 88–96.

[3] F. Baader, *The description logic handbook: theory, implementation, and applications*. Cambridge university press, 2003.

[4] D. Bamman, T. Underwood, and N. A. Smith, "A bayesian mixed effects model of literary character," in *Proceedings of the 52st Annual Meeting of the Association for Computational Linguistics (ACL14)*, 2014.

[5] K. Bontcheva, V. Tablan, D. Maynard, and H. Cunningham, "Evolving GATE to meet new challenges in language engineering," *Natural Language Engineering*, vol. 10, no. 3-4, pp. 349–373, 2004.

[6] D. K. Elson, N. Dames, and K. R. McKeown, "Extracting social networks from literary fiction," in *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 2010, pp. 138–147.

[7] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1535–1545.

[8] B. Fisseni, A. Kurji, and B. Löwe, "Annotating with Propp's morphology of the folktale: reproducibility and trainability," *Literary and Linguistic Computing*, vol. 29, no. 4, pp. 488–510, 2014.

[9] H. He, D. Barbosa, and G. Kondrak, "Identification of speakers in novels." in *ACL (1)*, 2013, pp. 1312–1320.

[10] M. Horridge and S. Bechhofer, "The OWL API: A Java API for OWL ontologies." *Semantic Web*, vol. 2, no. 1, pp. 11–21, 2011.

[11] N. Kazanina and C. Phillips, "Differential effects of constraints in the processing of Russian cataphora," *The Quarterly Journal of Experimental Psychology*, vol. 63, no. 2, pp. 371–400, 2010.

[12] R. Lang, "A declarative model for simple narratives," in *Proceedings of the AAAI fall symposium on narrative intelligence*, 1999, pp. 134–141.

[13] G.-M. Park, S.-H. Kim, and H.-G. Cho, "Structural analysis on social network constructed from characters in literature texts," *Journal of Computers*, vol. 8, no. 9, pp. 2442–2447, 2013.

[14] F. Peinado, P. Gervás, and B. Díaz-Agudo, "A description logic ontology for fairy tale generation," in *Procs. of the Workshop on Language Resources for Linguistic Creativity, LREC*, vol. 4, 2004, pp. 56–61.

[15] V. I. Propp, *Morphology of the Folktale*. American Folklore Society, 1958, vol. 9.

[16] N. Reiter, A. Frank, and O. Hellwig, "An NLP-based cross-document approach to narrative structure discovery," *Literary and Linguistic Computing*, vol. 29, no. 4, pp. 583–605, 2014.

[17] 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.

[18] D. Thakker, T. Osman, and P. Lakin, "Gate Jape grammar tutorial," *Nottingham Trent University, UK, Phil Lakin, UK, Version*, vol. 1, 2009.

[19] K. van Dalen-Oskam, J. de Does, M. Marx, I. Sijaranamual, K. Depuydt, B. Verheij, and V. Geirnaert, "Named entity recognition and resolution for literary studies."

[20] B. Varga and A. Groza, "Integrating DBpedia and SentiWordNet for a tourism recommender system," in *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 133–136.

[21] H. Xu, S. P. Stenner, S. Doan, K. B. Johnson, L. R. Waitman, and J. C. Denny, "Medex: a medication information extraction system for clinical narratives," *Journal of the American Medical Informatics Association*, vol. 17, no. 1, pp. 19–24, 2010.