Interleaving Ontology-Based Reasoning and Natural Language Processing for Character Identification in Folktales

Daniel Suciu and Adrian Groza*
*Intelligent System Group,
Department of Computer Science,
Technical University of Cluj-Napoca, Romania
Suciu.Daniel@isg.cs.utcluj.ro
Adrian.Groza@cs.utcluj.ro

***Abstract*—We propose a system for identifying literary characters from folktales. The process of extracting characters from free texts is guided by an ontology that encodes the knowledge of the folktale domain. We present how the ontology works with Natural Language Processing in GATE to increase the accuracy of character recognition. We validated the solution against various folktales.**

***Index Terms*—Natural language processing, ontologies, characters identification, folktales.**

I. Introduction

Information Extraction systems are developed to process and extract pieces of information from unstructured data that are of interest to the user. In this paper, we present a system that extracts information for the restrained world of folktales. The pieces of information we are looking for are represented by the literary characters and the unstructured data is the natural language in which the folktales are expressed.

The main approach in identifying literary characters from folktales is to use Natural Language Processing (NLP) techniques for defining syntactic patterns that help to extract the actors of the folktale. An enhancement of this approach is to develop an ontology that guides the extraction process. This ontology represents the knowledge component which the system uses to take the decision if the expression being analyzed is in fact a character [3]. In line with [3], we enhanced this new approach for obtaining a system that has good results for a small-size corpus of folktales.

The outline of the paper is the following: Section II introduces Description Logics and natural language processing as technical instrumentation used in our approach. In section III we specify the problem and address the conceptual difficulties. Section IV presents the developed computational model. Section V describes our experiments and the obtained results. Section VI discusses related work, while section VII summarizes the contributions.

II. Technical instrumentation

A. Description Logic

In the description logic $\mathcal{ALC}$, concepts are built using the set of constructors formed by negation, conjunction, disjunction, value restriction, and existential restriction [1] (Table I). Here, C and D represent concept descriptions, while r is a role name. The semantics is defined based on an interpretation $I = (\Delta^I, \cdot^I)$, where the domain Δ^I of I contains a non-empty set of individuals, and the interpretation function $\cdot^I$ maps each concept name C to a set of individuals $C^I \in \Delta^I$ and each role r to a binary relation $r^I \in \Delta^I \times \Delta^I$. The last column of Table I shows the extension of $\cdot^I$ for non-atomic concepts.

TABLE I
Syntax and semantics of $\mathcal{ALC}$.

Constructor	*Syntax*	*Semantics*
negation	$\neg C$	$\Delta^I \setminus C^I$
conjunction	$C \sqcap D$	$C^I \cap D^I$
disjunction	$C \sqcup D$	$C^I \cup D^I$
existential restriction	$\exists r.C$	$\{x \in \Delta^I \mid \exists y : (x,y) \in r^I \wedge y \in C^I\}$
value restriction	$\forall r.C$	$\{x \in \Delta^I \mid \forall y : (x,y) \in r^I \rightarrow y \in C^I\}$
individual assertion	$a : C$	$\{a\} \in C^I$
role assertion	$(a,b) : r$	$(a,b) \in r^I$

A terminology $TBox$ is a finite set of terminological axioms of the forms C $\equiv$ D or C $\sqsubseteq$ D. An assertional box $ABox$ is a finite set of concept assertions *(instance a C)* or role assertions *(related a b r)*, where C designates a concept, r a role, and a and b are two individuals. Usually, the unique name assumption holds within the same $ABox$.

A concept C is satisfied if there exists an interpretation I such that $C^I \neq \emptyset$. The concept D subsumes the concept C, represented by *(implies C D)* if $C^I \subseteq D^I$ for all interpretations I. Constraints on concepts (i.e. *disjoint*) or on roles (*domain*, *range*, *inverse* role, or *transitive* properties) can be specified in more expressive description logics[1].

B. Natural Language Processing

Natural Language Processing (NLP) involves solving complex task that appear frequently when working with free texts. For helping researchers and others to bootstrap their work, several NLP frameworks have been developed. One such

[1]We provide only some basic terminologies of description logics in this paper to make it self-contained. For a detailed explanation about families of description logics, the reader is referred to [1].

framework is the General Architecture For Text Engineering (GATE) framework. In GATE the logic is arranged in modules that are called pipelines. The output of one pipeline can be the input of another pipeline and several pipelines can be grouped to form a more complex system.

GATE framework contains an information extraction pipeline called ANNIE that is a good starting point for creating other custom systems. This pipeline is composed of several components: Tokenizer, Gazetter List, Sentence Splitter, POS Tagger, Semantinc Tagger that annotates entities such as Person, Organization, Location, and an Orthographic Coreference that adds identity relations between the entities annotated by the Semantic Tagger. For developing our system we have removed the last two components of the ANNIE pipeline and add a nominal phrase chuncker [8] and a syntactic parser [2]. These two new components are used by the semantic tagger we have developed for processing folktales.

Our semantic tagger is built using several JAPE grammar rules. A JAPE rule identifies a syntactic pattern in the text and annotates it with semantic information. The semantic information to be added to the annotation is obtained from an ontology that formalizes the domain of interest. We defined JAPE rules that determine if a nominal phrase is a candidate character, and a JAPE rule to determine if a nominal phrase is a character reference. For extracting relations between the identified characters, we constructed rules that check if the characters on which the relation is applied satisfy the imposed domain and range property.

III. Identifying characters in Free Texts

A. Problem Statement

The problem we address is that given a folktale, we identify all the nominal phrases from the text representing a character and then group these nominal phrases according to the character they belong to.

In the domain of folktales, this problem becomes more interesting because the characters are rarely specified by their names, but by the roles or functions they have. For example a character that is initially known as ‘a boy’ may become ‘a prince’ by marrying ‘a king’s daughter’. After the old king dies he might become ‘a king’. As the example shows the character has changed his role as the folktale evolves, moreover it has taken the role played by other character in a different part of the folktale, but the system must identify all those instances as representing the same character.

To be able to follow the transition of roles of a character, the system must have a knowledge powered component that understands how the world in which the characters interact works. In other words, we need a component that enables us to encode the logic that governs the world of the characters. Our solution is based on Description Logic.

B. Problem Analysis

The starting point for investigating how characters can be identified in narrative texts is given by the observation made by [4] who states that indefinite nominal phrases have the role of introducing new characters in the discourse of a text. Then these indefinite nominal phrases can be referred back in further sections of the text by definite nominal phrases. By using this principle, we can model the occurrences of a character throughout the text as a set composed of an indefinite nominal phrase and its corresponding definite nominal phrases.

We start by defining some terms that are used in this paper. A candidate character is an indefinite nominal phrase that represents a character. A character reference is a definite nominal phrase that represents a character and it is determined by a candidate character, i.e. it refers to a candidate character. In the rest of this section we describe the main phases of our system, followed by a description of the solutions we found for specific sub-problems and how are these solutions integrated in the main workflow.

The developed system is divided into five phases that are executed sequentially (see Fig. 1).

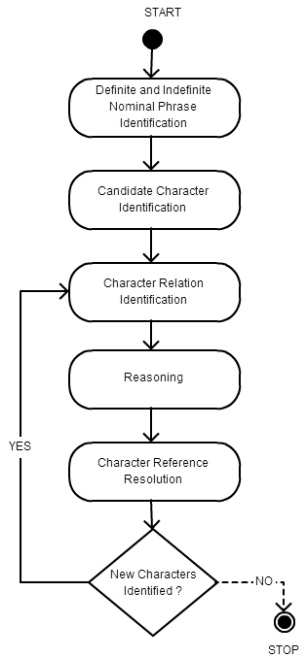

Fig. 1. Main execution phases.

The first phase of our system consists in identifying all the indefinite and definite NP from the text. An indefinite NP is considered to be a nominal phrase that has as determinants either ‘a’, ‘an’ or ‘some’, while a definite nominal phrase has as determinant the term ‘the’. This phase and the next are the only ones that are executed only once, they are not part of the set of phases that are executed iteratively.

In the second phase, we analyze which of the previously found indefinite nominal phases are characters, as according to [4], indefinite NPs are the means used by authors to introduce new characters in a narrative text. For making this decision we use an ontology that contains the definitions of characters we are trying to identify. Such an ontology contains definitions as: mother is a subtype of woman, daughter is subtype of girl, both are subtypes of person; or more complex definitions such as: a sister is a girl that has the same parents as another

person. To complete the second phase of the system we are comparing the head of the NPs with the names of the classes of the ontology and when there is a match we annotate that NP as being a candidate character and add its head term as an individual of that particular class. At the end of this phase, the ontology contains all the characters introduced by the author using indefinite NP.

The third phase extracts the relations between the identified candidate characters. For this we have defined a number of relations in the ontology. We focused on defining relations that describe how the members of a family interact with each other, therefore we have relations such as: has child, has parent, has brother, has sister, etc. The ontology also contains relations that describe the social status of a character: becomes king, becomes queen. Each relation in the ontology is annotated with one or multiple expressions that could represent it in a free text. We also define for each relation the domain and range constraints, so that it cannot be applied between characters that have no semantic meaning together. Then we search the text to identify all these expressions. When such an expression is found, we check to see if it relates two identified characters. If it is the case, we use the domain and range properties of the relation to determine if the two characters respect the imposed constraints. If they do respect the relation's constraints, then we add the relation to the ontology.

After the candidate characters and the relations between these characters have been extracted and added into the ontology, the fourth phase starts. In this phase we use a reasoner to deduce new information. The purpose of using a reasoner at this step is to discover if a certain character can be cast in other roles than the ones that were identified.

Example 1. *The folktale 'The Magic Swan-Geese' starts with the following phrase: 'Once upon a time a man and a woman lived with a daughter and small son.'*

From this phrase, in the first step we identify 'a man', 'a woman', 'a daughter' and 'small son' as candidate characters. In case of 'small son' we use coordinate conjunction propagation of the determinant 'a' to determine that the NP is also a candidate character. Then we identify the expression 'lived with' as a possible instance of the relation 'hasChild' defined in the ontology (see figure describing ontology's object properties in section IV-C). Next, we search for the characters on which to apply the relation and find that the character 'a woman' has a relation of type 'hasChild' with the character 'a daughter'. Before adding this relation to the ontology, we perform an additional verification step that consists in checking if the characters are of the type specified in the domain and range property of the relation. If they respect the domain and range constraints, we add the relation to the ontology. Then we submit the ontology to a reasoner. In this case the reasoner deduces that the man is also a father, the woman a mother, the daughter a sister and the son a brother.

After the reasoner completes the inference process and updates the ontology with the newly obtained information, the fifth phase begins: character reference resolution. Now we have to determine for each definite nominal phrase if it refers to one of the identified candidate characters. To do this we work closely with the developed ontology. For each definite nominal phrase we take its head term and compare it to the class names defined inside the ontology. If we find a match, then it means that this definite NP is a character and we must find if it refers to one of the candidate characters. So we check the ontology to see if the matched class contains individuals that are candidate characters and were identified in the text before the definite NP being analyzed. Here the problem becomes more complex, depending on how many individuals the class has that respect the two conditions mentioned before. There exists three cases:

- The class contains no individuals. That means that until this iteration we have not found any candidate character of this type.
- The class contains only one individual, meaning that this definite NP refers to that individual, i.e. this definite NP is a character reference that refers back to the candidate character represented by the individual.
- The class contains more than one individual. Here we have to find a metric that determines to which of the individuals the definite NP refers to. We have chosen a simple but effective metric called 'Closest Individual' metric. This metric says that a definite NP refers to the closest character that is defined before him. This metric also solves the case when the same role/class, for example 'a king', is used in different parts of the folktale to introduce different characters.

This is the final phase of an iteration. If there were new characters identified in this iteration then we perform another iteration, otherwise we stop because we cannot identify any other characters. A further iteration is not going to execute the nominal phrase identification and candidate character identification phase, it starts directly from the relation identification phase. The reason for doing this is obvious, we cannot find any other candidate characters no matter what kind of information we extract during an iteration, but we can find new relations based of the new character reference we are able to solve.

The phases presented up to this point represent the backbone structure of our system. The evaluation of the system on our text corpus revealed that certain types of characters were not discovered. To improve the performance of our system we developed some enhancements to the system that are listed below and then described in more details in the remaining of this section. The developed enhancements are:

1) System identifies characters introduced by definite NP
2) System identifies NPs that have implicit definite and indefinite articles as characters
3) A identified relation is applied on all the members of a coordinate conjunction
4) System identifies characters representing a group of persons

The first enhancement addresses the problem of characters that are introduced by definite NP. In the folktale domain, there

exists some well-known characters that are considered to be familiar to the reader and therefore are not introduced in the discourse of the text by using an indefinite NP. These types of characters are not recognized by the backbone solution, because it considers that all characters must be introduced by an indefinite NP. To solve this kind of situations we add the following logic to the main solution: after all the iterations have finished we start processing the definite nominal phrases again. This time we are searching for the first definite nominal phrase that matches a class defined inside the ontology that has no individual added to it. When such a definite NP is found, it is annotated as a 'virtual' candidate character and a new iteration is started. In this new iteration the found definite NP is treated as any other candidate character. The algorithm re-evaluates every previously found character references to determine if the candidate character they currently refer should not be changed to the newly added candidate character. The idea behind this approach is that if after the system is done processing the text, we are still able to find definite nominal phrases that according to our ontology represent characters then it means that either the respective characters were not introduced by indefinite nominal phrases or the system was not able to detect them. Thus better than not identifying the characters occurrences at all, we start identifying them from this point further.

Example 2. *In the folktale 'The Magic Swan-Geese' one character, the witch, is introduced in the following way: 'In the cabin was the old witch Baba Yaga, spinning flax.' Then during the text she is further referred as: 'the old witch' and 'the old woman'.*

In example 2, the witch character is not identified by the main solution, because it isn't introduced by an indefinite NP. However, before the solution finishes its execution, the newly added logic discovers that there is a definite NP 'the old witch' whose head term 'witch' matches one of the ontology's class that has no individual. So the noun phrase 'the old witch' is marked as a candidate character and a new iteration is started. We mention that in this particular case the algorithm doesn't work perfectly, because it marks as candidate character the second definite NP 'the old witch' and not the first one 'the old witch Baba Yaga'. This is caused by the fact that it considers as head term of the NP 'the old witch Baba Yaga' the term 'Baba'. Then in the newly started iteration, for the character 'the old woman', the algorithm changes the candidate character it initially refers, from the character 'a mother' (presented at the beginning of the the text) to the character 'the witch', correcting in this way a misclassification. This happens because the witch is also a woman, and it is closer to the analyzed character then the mother.

The second enhancement approaches the problem of nominal phrases that have no explicit articles. In case of coordinate conjunctions not all members have an article, because it is considered that they all have the same article as their first member. This represents a problem to our solution, because in the process of identifying characters it only considers nominal phrases that have an article. To solve this issue we use a syntactic parser that identifies dependencies between nominal phrases. For each indefinite and definite NP that is related to other NP by a coordinating conjunction of type 'and' or 'or', we explicitly propagated its article to all the members of the conjunction.

Example 3. *In the folktale 'The Magic Swan-Geese' we have the sentence: 'The father and mother went off to work'.*

In example 3, the mother character is not initially discovered, because the NP 'mother' is not considered a definite NP. However with the propagation of articles through coordinating conjunction, the system is able to identify the NP 'mother' as a definite one and then as a character reference.

The third enhancement solves the case of applying a relation on all the literary character members of a 'and' or 'or' coordinate conjunction. When a relation between two characters is identified the system must detect if those character are in a coordinate conjunction with other characters. If it is the case, then the relation must be applied on all members otherwise we are not extracting all the knowledge from the text. This problem was solved by using syntactic dependencies between nominal phrases, allowing us to determine the members of a coordinate conjunction.

Example 4. *In the folktale 'The Magic Swan-Geese' we have the sentence: 'Once upon a time a man and a woman lived with a daughter and small son.'*

In example 4, the Stanford typed dependencies parser [2] generates the dependencies presented in Fig. 2.

```
advmod(lived-10, Once-1)
prep(Once-1, upon-2)
det(time-4, a-3)
pobj(upon-2, time-4)
det(man-6, a-5)
nsubj(lived-10, man-6)
cc(man-6, and-7)
det(woman-9, a-8)
conj(man-6, woman-9)
root(ROOT-0, lived-10)
prep(lived-10, with-11)
det(daughter-13, a-12)
pobj(with-11, daughter-13)
cc(daughter-13, and-14)
amod(son-16, small-15)
conj(daughter-13, son-16)
```

Fig. 2. Typed dependencies of sentence in example **??**.

From Fig. 2, the characters man (token 6) and woman (token 9) are related trough a coordinate conjunction *conj(man-6, woman-9)*, while daughter (token 13) and son (token 16) are also related through a coordinate conjunction *conj(daughter-13, son-16)*. The results also show that the character daughter is determined by an indefinite article 'a' (token 12) *det(daughter-13, a-12)*.

For this example, the system determines that the characters 'a man' and 'a woman' are connected by a 'and' conjunction and that the characters 'a daughter' and 'small son' are also

connected by a 'and' conjunction. Thus the system identifies four relations of type 'hasChild': i) 'a man' hasChild 'a daughter'; ii) 'a man' hasChild 'small son'; iii) 'a woman' hasChild 'a daughter'; iv) 'a woman' hasChild 'small son'.

From this four relations, the system can deduce the following facts about the characters:

1) The character 'a man' is also a 'Father', because he is of type 'Man' and has at least one relation of type 'hasChild' with another 'Person'.
2) The character 'a woman' is also a 'Mother', because she is of type 'Woman' and has at least one relation of type 'hasChild' with another 'Person'.
3) The character 'a daughter' is also a 'Sister', because she is a 'Girl' and has at least one relation 'hasParent' (which is the inverse relation of 'hasChild') with another 'Person' that has another child besides her.
4) The character 'small son' is also a 'Brother', because he is a 'Boy' and has at least one relation 'hasParent' with a 'Person' that has another child besides him.

The fourth enhancement is regarding the identification of characters that represent a group. The current solution does not determine the individual characters that belong to a group, but it sees the group as a distinct character in the folktale. In the ontology, we defined for each class, where it makes sense, an equivalent class that represents a group.

Example 5. *In the ontology we defined for the class 'girl' a group class 'girls', for 'boy' a group class 'boys', for 'woman' a group class 'women' and the same for other classes.*

Then there appeared a new problem in the character reference resolution step, when a group character reference referred back to a single candidate character. Thus we modified the character reference step so that it takes into consideration if the character represent a single person or multiple persons, allowing multiple person character reference to refer only to multiple person candidate characters and the same for single person character reference.

IV. Proposed method

A. System Architecture

The system we have developed for identifying literary characters from folktales is composed out of three major components. The first component, called Text Processing Component, performs all the Natural Language Processing transformations that are needed for extracting from the input text a set of entities that can be manipulated by the system. In this case we are interested in nominal phrases that can describe an actor of the folktale.

The second component, Character Identification Component, uses the nominal phrases obtained by the NLP component and tries to identify which of these nominal phrases represent characters and what are the relations between the identified characters. In order to achieve its goal, it uses the third major component of the system, where all the knowledge about the folktale's world is encoded. Before explaining in more detail how the components interact, we present an overview of the system architecture.

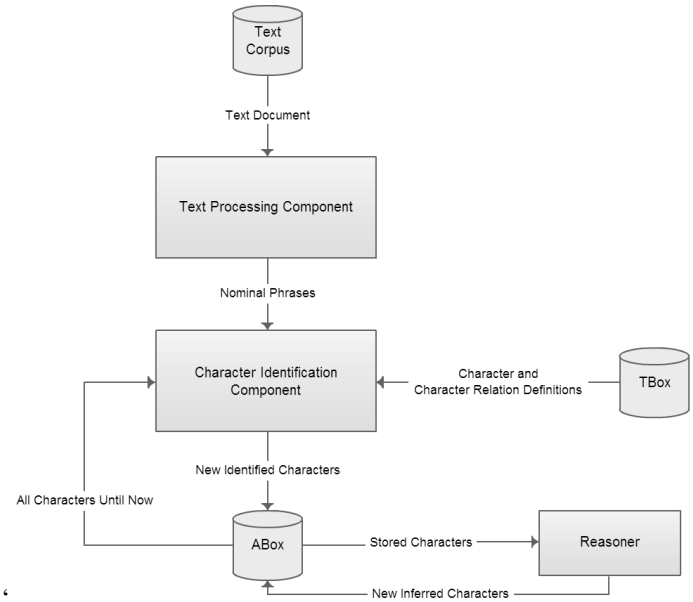

Fig. 3. Overview of system architecture

In Fig. 3, the input of the system is processed by the Text Processing Component. This component applies a sequence of processes to extract the nominal phrases from the text. It is built on top of the ANNIE Pipeline, which is integrated in the GATE framework. The pipeline performs sequentially the following processes: it divides the input text into tokens using a Tokenizer, then it splits the text into sentences, tags the tokens with the corresponding part of speech (POS) and finally it uses the nominal phrase chuncker provided by [8] to determine the nominal phrases of the text. Because the system also needs to study the relation between the nominal phrases, we also use Stanford dependency parser [2] to generate the dependencies between the tokens of the text. The obtained dependencies are then used by the Character Identification Component.

The nominal phrases obtained by the previous component are then given as input to the Character Identification Component, which is build out of several GATE pipelines. Each pipeline has a set of custom defined JAPE grammar rules used to identify the needed linguistic patterns from the text. We have one pipeline that is responsible for identifying the candidate characters from the text and inserting them into an ontology. Another pipeline accomplishes the task of extracting relation instances between the already known characters. A third pipeline finds character references to the candidate characters that were previously identified.

All these pipelines work closely with the TBox we have defined, which is implemented using an OWL ontology. This ontology focuses on capturing the relations that exists between the family members and between characters that have different social roles in the folktale world. We use the ontology to have

the needed knowledge when we must decide if a particular nominal phrase is a character. If we decide that a nominal phrase is indeed a character then we insert it into the ABox. Then the assertions from the ABox and the definitions from the TBox are used by the Reasoner to infer new information about the known characters. If the Reasoner is able to obtain new information about some of the identified characters, then a new iteration is performed. From the presented system architecture, the Reasoner component is the only one that is not implemented by us. As the implementation for the Reasoner module, we chose Pellet [10].

B. Annotation process

In this section we describe the techniques used by the system and the human annotator to annotate characters in the text corpus. The main objective of the annotation process is to identify all the nominal phrases that represent a character. We consider for annotation only definite and indefinite nominal phrases. The second objective of the annotation process is to enable the creation of a system that automatically evaluates the system's results compared with a manually annotated corpus. The last objective is to provide an encoding of the identified characters so that they can be easily recognized inside the ontology.

The last objective must be satisfied only by the system. It adds the following two features to our LiteraryCharacter annotation: CH_ID and A_CH_ID. These two features must always be interpreted together. They have the following meaning: if the nominal phrase being analyzed is a candidate character then CH_ID represents the id of this character and A_CH_ID is not set. If the NP is a character reference then A_CH_ID (where A comes from anaphoric) represents the character id and CH_ID represents the id of the candidate character it refers. Each feature is incremented separately when a new candidate character or character reference is found.

Example 6. *Consider the following to be part of the annotation of a definite NP that is a literary character:* {*CH_ID=CH2, A_CH_ID= A_CH5*}

The interpretation is the following: this definite NP is the fifth character reference found in the text and it refers back to the second candidate character found.

To satisfy the second objective, two similar features to the ones previously presented are added to the annotation: GLOBAL_ID and A_GLOBAL_ID. They also must be interpreted together and have the same meaning as the ones already described. The difference is that they are not incremented when a new character is found, but are preprocessed at the beginning of the system. Thus a NP always has the same GLOBAL_ID, no matter of how many characters we find. This allows us to use it for comparing the system's results to the manually annotated corpus.

The last feature contained by a LiteraryCharacer annotation is the HEAD feature. This features represents the head term of the nominal phrase and therefore the character name.

An annotation from the manually annotated corpus is considered to be the same as one generated by the system, if they cover the same text span and have the GLOBAL_ID, A_GLOBAL_ID and HEAD features equal.

C. Engineering the folktale ontology

One of the key component of our system is represented by the folktale ontology we have developed. For determining what are the concepts and relations that should be modeled by the ontology, we implemented a system that generates a list of the most common terms found in our text corpus. By analyzing the obtained list we discovered that our ontology should focus on three types of concepts. The most important concepts were the ones that describe the relationships between family members, followed by the concepts describing the social status of a character and the concepts modeling the animal and supernatural world.

Before starting to have a more detailed look of the ontology, we must state that the goal of this ontology is to cover both single person characters and multiple person characters, i.e. groups of persons. In this line, we have considered for some classes both the single and multiple form. For example, the class 'Person' has two subtypes 'SinglePerson' denoting one person and 'MultiplePerson' denoting a group of persons. The same pattern is also applied for other classes. When presenting the concepts of the ontology we discuss only about the ones in the single category. We start by describing the family-related concepts of our ontology that are presented in Fig. 4.

1. Man $\sqcup$ Woman $\sqcup$ Boy $\sqcup$ Girl	$\sqsubseteq$	SinglePerson
5. Father	$\equiv$	Man $\sqcap$ $\exists$hasChild.Child
6. Mother	$\equiv$	Woman $\sqcap$ $\exists$hasChild.Child
7. Son	$\equiv$	Boy $\sqcap$ $\exists$hasParent.Parent
8. Daughter	$\equiv$	Girl $\sqcap$ $\exists$hasParent.Parent
9. Parent	$\equiv$	Father $\sqcup$ Mother
10. Child	$\equiv$	Son $\sqcup$ Daughter
11. Sibling	$\equiv$	Brother $\sqcup$ Sister
12. Brother	$\equiv$	Sibling $\sqcap$ Boy
13. Sister	$\equiv$	Sibling $\sqcap$ Girl
14. Consort	$\equiv$	Husband $\sqcup$ Wife
15. SingleFamilyMember	$\equiv$	Parent $\sqcup$ Child $\sqcup$ Consort $\sqcup$ Sibling $\sqcup$ GrandParent
16. SingleFamilyMember	$\sqsubseteq$	SinglePerson
17. SinglePerson	$\sqsubseteq$	Person

Fig. 4. Family-related concepts in the folktale ontology

Our ontology models a human person as being of four types: a man, woman, girl or boy. By using this four fundamental classes we construct the concepts describing a family. Thus the concept of father is modeled as a man that has a relation of type hasChild with at least one child in the folktale. The same applies for the mother with the modification that it must be of type woman. A son is represented as a boy that has a relation of type hasParent with at least one other person. A sister is modeled as a person who is in the same time a sibling and also a girl.

The concepts from Fig. 4 are then used to describe the social status of a character in the folktale. A part of these concepts are presented in Fig. 5.

1. SocialStatus $\equiv$ SingleSocialStatus $\sqcup$ MultipleSocialStatus
2. King $\sqcup$ Queen $\sqsubseteq$ SingleSocialStatus
4. Prince $\equiv$ Boy $\sqcap$ ($\exists$hasParent.King $\sqcup$ $\exists$hasParent.Queen)
5. Princess $\equiv$ Girl $\sqcap$ ($\exists$hasParent.King $\sqcup$ $\exists$hasParent.Queen)
6. Witch $\equiv$ SingleSocialStatus $\sqcap$ Woman $\sqcap$ Enchantress

Fig. 5. Social status related concepts in the folktale ontology

The King or a Queen concepts are modeled just as subconcepts of SingleSocialStatus. A prince/princess is represented as a boy/girl that has at least one relation hasParent to another individual of type king or queen. In Fig. 5 there is a difference between the definition of a witch and that of a queen, where a witch is restrained to be a woman while a queen not. This difference comes as a result of performing some experiments with the ontology on our text corpus. We observed that by restraining the individuals of the concept queen to the ones in the concept of woman the performance of our system decreases. This is caused by the fact that in the world of folktale a queen is never referred to as a woman and by making the queen a subtype of woman, some of the woman characters were incorrectly classified by the system.

The last set of concepts we are discussing is the one related to the animal and supernatural creature (Fig. IV-C). The animal class has two subclasses, SingleAnimal used to represent a one individual animal and MultipleAnimal used to represent a group of animals. The same applies for the Supernatural class.

2. Animal $\equiv$ SingleAnimal $\sqcup$ MultipleAnimal
3. Frog $\sqcup$ Horse $\sqcup$ Mouse $\sqsubseteq$ SingleAnimal
7. Supernatural $\equiv$ SingleSupernatural $\sqcup$ MultipleSupernat
8. Giant $\sqsubseteq$ SingleSupernatural

Fig. 6. Animal and supernatural related concepts in the folktale ontology

In Fig. 7 we enumerated some of the relations that are used to relate the concepts described before. The 'hasChild' relation has as inverse relation the 'hasParent' relation and two subrelations: 'hasDaughter' and 'hasSon', describing the relation between a parent and a daughter or son. Another relation is the 'hasConsort', which models the relation between a husband and his wife. It has two more specific subrelations: 'hasWife' and 'hasHusband'. These two subrelations are functional, meaning that one individual can have only one such relation with another individual and they are also inverse function.

hasChild $\equiv$ hasParent $^-$	hasDaughter $\sqsubseteq$ hasChild
Domain: SinglePerson	Domain: SinglePerson
Range: SinglePerson	Range: Daughter
hasSon $\sqsubseteq$ hasChild	hasConsort $\sqsubseteq \top$
Domain: SinglePerson	Domain: SinglePerson
Range: Son	Range: SinglePerson
hasWife $\sqsubseteq$ hasConsort	hasHusband $\sqsubseteq$ hasConsort
hasWife $\equiv$ hasHusband$^-$	hasHusband $\equiv$ hasWife$^-$
Domain: SinglePerson	Domain: SinglePerson
Range: Wife	Range: Husband
Characteristics: functional	Characteristics: functional

Fig. 7. Object properties

The ontology we have described is populated by the system with the CH_ID of a candidate character or with the A_CH_ID in case we have a character reference. For populating the ontology we have used both the Ontology API provided by GATE framework and the OWL API [5].

V. Evaluation And Results

In order to evaluate the performance of our system we have manually annotated seven folktales, having in total 453 annotations. We then compared these annotations with the ones generated by the system and the results are shown in table V.

TABLE II
Results for evaluated folktales.

Folktale	*# Ann.*	*Recall*	*Precision*
The Magic Swan-Geese	60	0.75	0.98
The Frog King	37	0.70	0.72
The King's Son Who Feared Nothing	69	0.81	0.82
Faithful John	102	0.62	0.72
The Twelve Brothers	57	0.68	0.74
Rapunzel	41	0.78	0.74
The Three Little Men in the Wood	87	0.68	0.79
Total	*453*	-	-
Average	*65*	*0.72*	*0.79*

The results show how many character occurrences have been correctly identified and not how many characters out of the total were found. For example, in the case of the folktale 'The Magic Swan-Geese', from the 60 annotations that represent a character the system has identified 75% of them and 98% out of those annotation are correct.

For the folktale 'The Magic Swan-Geese' the system has found 9 characters out of 11. The character a man is found in two places as 'a man' and 'the father'. The character a woman is found in three places as 'a woman' and 'the mother'. The character 'a daughter' is found in 22 places as 'a daughter, 'the daughter', 'the sister', 'a good girl' and 18 times as 'the girl'. The system also identified the witch character, even though it is not introduced by an indefinite nominal phrase. The witch character is found in two places as 'the old witch' and 'the old woman'. One problem the system has with this folktale is that it cannot identify one of the main characters 'the Swan-Geese', because the noun phrase chuncker [8] does not recognize 'swan-geese' as a nominal phrase. Another problem is that the character 'a daughter' is actually identified as two characters: one candidate character 'a daughter' with its character references 'the daughter' and 'the sister' and one candidate character 'a good girl' with 18 character references 'the girl'. This could be solved in a post-processing step, where we analyze all candidate characters to see if some of them could be merged to form only one character. We take the decision of merging two characters when we observe that they have a relation in common that make them more probably to be the same character than two separate character. Such a relation could be that two characters have the same parent.

In case of the folktale 'The Frog King' we identify all the characters, except the character servant Henry, because the system doesn't know how to treat characters that are specified by name. Another problem is that the frog character

is introduced several time throughout the text as ‘a frog’ and the system identifies 4 frog characters instead of one. The system also doesn’t identify the transformation of the frog into a king, so in the last part of the folktale where the character references ‘the king’ actually refers to the frog character, the system interprets these character references as referring to the king character introduce at the beginning of the text.

For the folktale ‘The Twelve Brothers’, the system finds the character of the twelve boys, although they are not introduced by an indefinite NP. Moreover, the system identifies the characters ‘the twelve brothers’ and ‘the twelve boys’ as being the same character. A first problem that should be solved is the one when a character is introduced more than once using the same title (‘a king’s son’) or using different titles (‘a king’s son’ and then as ‘a man’). Another problem that should be solved is the one of possessive pronouns, adjectives or nouns. For example, the system increases its performance if it can distinguish between the characters ‘the man’s daughter’ and ‘the woman’s daughter’. At this point, the system sees both characters as being the same character.

VI. Discussion and Related Work

The domain of narrative texts have been studied constantly, first from a linguistic point of view and then, based on these linguistic views, from a computational perspective. In case of the folktale subdomain, there have been proposed several theories even before the era of digital information. The theory proposed by the russian folklorist and scholar Vladimir Propp in [9] states that characters of a folktale can be grouped into seven categories and that the narrative of the folktale can be constructed from 31 narrative functions.

This structured description of a folktale enabled various researchers to build computation models that are capable of extracting different kind of features from a folktale. An interesting model developed by [7] identifies characters of the folktale and the actions performed by them. The work done by [6] identifies character and then ranks them according to their importance. In line with [3], we model the knowledge that describes the folktale world using an ontology to guide the character identification process. Differently from [3], we focus on literary characters that are not introduced by an indefinite nominal phase. We have also identified characters that represent a group of persons and not a single individual.

A minimal number of syntactic patterns have been used in [7] to identify text spans that can be transformed to character or actions. The analysis shows that interactions between actors are typically represented by the pattern NP+VP(+NP). They also developed a procedure that assigns semantic markup to text string representing characters. In our case, the central component of the system is represented by the ontology component, who operates at the semantic level and not at the syntactic level. We have used only one syntactic pattern, NP+VP+NP, to identify relations between literary characters.

The work presented in [6] extracts the actors of a folktale by starting from the idea that a character differentiates itself from other entities of the folktale by expressing intentionality or consciousness. There are two important linguistic constructs for expressing these attitudes, through direct and indirect speech. Moreover, they determine the importance of characters by computing their dispersion in the folktale. For this they have developed two sets of syntactic patterns that are able to identify direct and indirect speeches in a text. This idea of determining the importance of a character based on its dispersion in text can be easily integrated within a future enhancement of our proposed system.

VII. Conclusions

We showed that accurate results can be obtained by developing and integrating an ontology in the process of identifying literary characters from folktales. Even though we have focused only on one syntactic relation, between indefinite nominal phrases and definite nominal phrases, we were able to discover characters with a high recall and precision. This approach also proved to be flexible as further enhancements of the extraction process can be added with minimal modifications regarding the knowledge component.

We are currently focusing on solving the problem of characters that are introduced more than once in the folktale and how we can unify them to form a single character. A second line of investigation is how possessive pronouns, adjectives and nouns can be used when performing character reference resolution to obtain better accuracy.

Acknowledgments

We thank the reviewers or their valuable comments. Adrian Groza is supported by the Technical University of Cluj-Napoca, Romania, through the internal research project ”GREEN-VANETS: Improving transportation using Car-2-X communication and multi-agent systems”.

References

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

[2] M.-C. de Marnee and C. D. Manning. (2008) Stanford typed dependencies manual. [Online]. Available: http://http://nlp.stanford.edu/software/dependencies_manual.pdf

[3] T. Declerck, N. Koleva, and H.-U. Krieger, “Ontology-Based Incremental Annotation of Characters in Folktales,” in *6th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2012)*, Stroudsburg, PA, 24-24 April 2012, pp. 30–34, hU.

[4] K. v. Heusinger and U. Egli, *Reference and Anaphoric Relations*. Kluwer Academic Publishers, 2000.

[5] M. Horridge and S. Bechhofer, “The OWL API: A Java API for Working with OWL 2 Ontologies,” in *6th OWL Experienced and Directions Workshop*, Chantilly, Virginia, october 2009.

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

[7] P. Lendvai, T. Váradi, S. Darányi, and T. Declerck, “Assignment of character and action types in folk tales,” *Nooj*, p. 102, 2010.

[8] (2012) MuNPEx website. [Online]. Available: http://www.semanticsoftware.info/munpex

[9] V. Propp, *Morphology of the Folktale: Revised and Edited with Preface by Louis A. Wagner, Introduction by Alan Dundes*. University of Texas Press, 2010, vol. 9.

[10] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, “Pellet: A practical OWL-DL reasoner,” *Web Semantics: science, services and agents on the World Wide Web*, vol. 5, no. 2, pp. 51–53, 2007.