Automatic Identification of Protagonist in Fairy Tales Using Verb

Hui-Ngo Goh, Lay-Ki Soon, and Su-Cheng Haw

Faculty of Computing and Informatics, Multimedia University, Jalan Multimedia, 63100 Cyberjaya Selangor, Malaysia
{hngoh,lksoon,sucheng}@mmu.edu.my

Abstract. Named entity recognition (NER) has been a well-studied problem in the area of text mining for locating atomic element into predefined categories, where "name of people" is one of the most commonly studied categories. Numerous new NER techniques have been unfolded to accommodate the needs of the application developed. However, most research works carried out focused on non-fiction domain. Fiction domain exhibits complexity and uncertainty in locating protagonist as it represents name of person in a diverse spectrums, ranging from living things (animals, plants, person) to non-living things (vehicle, furniture). This paper proposes automated protagonist identification in fiction domain, particularly in fairy tales. Verb has been used as a determinant in substantiating the existence of protagonist with the assistance of WordNet. The experimental results show that it is viable to use verb in identifying named entity, particularly "people" category and it can be applied in a small text size environment.

Keywords: Named entity recognition, characters, fairy tales, text mining.

1 Introduction and Motivation

Named entity recognition (NER) is a well-studied research in the area of information extraction (IE) aiming to locate and extract significant atomic elements in text into predefined categories. The most common studied categories are "name of people, organization and location" [1], [2] and [3], "date, time and phone" [4], "name of person, diploma, organization and research" [5] and many more entities as of interest of the application intended to be built.

Andrew e*t al.* used list of features, lexicon (augment using web search engine) and conditional random fields (CRF), a machine learning probabilistic approach to extract named entities in structured texts of CoNLL03 (name of person, location, organization and miscellaneous) [2]. Einat *et al.* focused solely in extracting personal names from informal text (email) using CRF and dictionary to enhance the names extraction [6]. The performance results vary among the chosen email corpora due to the free writing style in informal text and insufficient training data to produce good model for NER. Satoshi *et al.* manually hand-crafted about 1400 rules and 130,000 instances of dictionary to extract 200 categories of named entity covering generally Japanese newspaper domain [7]. In 2010, Laura *et al.* proposed domain

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 395–406, 2012.

adaptation of rule-based annotator to enhance domain customization for NER, a domain-independent CoreNER library of 104 features definition were being crafted manually to tailor different application domains need [1]. Public datasets of CoNLL03, Enron and ACE05 were used to train and test the "person, location and organization" entities. However, it is still manual and time consuming.

Character level model [3] used Hidden Markov Model (HMM) and Maximum-entropy Conditional Markov Model to inspect each letter in identifying named entity in ConLL, the character emission model is based on the *n*-gram proper-name classification engine [8] and state transition chaining is used to identify named entity boundary and classify them into predefined categories. Le *et al.* studied the use of inductive logic programming to extract named entities (name, diploma, organization, research) in Vietnamese language [5]. 80 Vietnamese homepages of scientist that were tagged manually and a set of features were used to train to generate a set of extraction rules to extract named entities in chosen test corpus. Javier *et al.* discussed the impact of coverage, reliability and independent number of features in extracting name of person [9]. Machine-learning algorithm, NER and classification were used to studied the mentioned impact, in the case of NER, combination of Stanford NE Recogniser (machine learning) and OAK (rule based English analyzer) were used to detect NE. It generates good performance results if all NE features are being used in the training process when producing trained model for NE recognition. Michael *et al.* studied further details of breaking down "name of person" into sub-categories such as "politician" and "entertainer", topic signatures and Wordnet are used to enhance the trained model using supervised machine learning [10].

All the above mentioned NER techniques are mainly constrained by two major issues as discussed below:

(1) *Recognition approach*: The evolutionary of NER begins with manual effort to semi-automatic, and then to automatic approaches. Manual NER requires excessive amount of time and resources from domain expert and knowledge engineer to hand-coded the rules manually. In such approach, existing text mining resources such as WordNet and dictionaries are always in used to speed up the manual NER process [7]. Knowing that manual construction of NER rules exhibits a promising performance results due to the intentionality of recognizing NE in the domain studied, Laura *et al.* explored the domain customization using rule-based annotator; a set of universal rules is needed to accommodate investigated domain needs [1]. However, flexibility and scalability is still the main issue to be dissolved in manual NER. Semi-automatic NER begins with seed (manual selection) NE. Often, machine learning is used to train the seed NE to generate a model for NER. The quality of chosen seed data will greatly impact the trained model for NE recognition. Automatic NER implies fully recognition without human intervention. It is not an easy task as each domain exhibits differently in term of context and structural text representation.

(2) *Nature of the domain*: Research domain done in the area of NER can be classified into fiction based and non-fiction based. Fiction implies literary work which is based on imagination and not necessary on facts, e.g. novel and fairy tales, whereas non-fiction denotes representation of a subject which is presented as fact, such as

manual, news-wired and tourism website. Most NERs developed are in non-fiction based. Non-fiction based exhibits certain patterns in identifying NE, for instance, name of person may start with designator, capital letter of the first character, naming in a human way. However, fiction based exhibits complexity and uncertainty in locating NER as it represents name of person in a diverse spectrums, ranging from living things (animals, plants, person) to non-living things (vehicle, furniture). Elson *et al.* employed quoted speech attribution (dialogue and internal monologue) and syntactical approach (adaptation of natural language tools) to identify character in literary fiction, specifically 19th century novels and serials [11]. However, its characters are represented in human alike name.

In this paper, we propose a fully automated named entity recognition framework to overcome the above mentioned issues. Fiction-based domain is used to test on the proposed framework. We study the predefined category of "name of person" but aim to recognize protagonist(s) in fairy tales. Stanford parser and Stanford dependency relation are used to shallowly parse the input file to extract potential NE from the natural text. Word(s) that is/are labeled as VERB between two potential NEs will be extracted to form syntactic triplet structure of subject – verb – object (S-V-O) at a sentence level. WORDNET is then used to substantiate the extracted verb that associates with human action in identifying protagonist. Finally, threshold value is used to filter potential NEs in locating protagonist(s). Part of the work of this paper is a replication and extension of previous research on ontology construction in fiction-based domain [12].

The outline of this paper is as follow: Section 2 discusses the technologies background for this work. Section 3 presents the proposed system framework. Section 4 describes the experiments and the paper is concluded in Section 5.

2 Technologies Background

2.1 Stanford Parser

Stanford parser is a probabilistic parser that analyses syntactic structure of natural language sentences. It has the performance of 86.36% of accuracy in parsing [13]. It is implemented in Java by Stanford University's Natural Language Processing Group and it is available in four languages (English, Chinese, Arabic and German). In this project, English is used to run and test on the selected plain text input.

The parser can read various forms of plain text input and return various analysis formats, including part-of-speech tagged text, phrase structure trees, and a grammatical relations (typed dependency) format. In this work, only phrase structure trees and grammatical relations are used.

2.1.1 Phrase Structure Parse

Phrase structure trees utilized unlexicalized probabilistic context free grammar (PCFG) to achieve greater efficiency and accuracy in parsing sentences. Generally, phrase structure tree is a syntactical structure of sentence that segment group of words into phrases to form the subject and object of the verb.

2.1.2 Stanford Dependencies

Stanford dependencies of English sentences are brought forth based on rules / patterns and Treebank representation from the generated phrase structure trees. There are forty-eight grammatical relations altogether to form the Stanford dependencies. It is a predicate argument representation with a grammatical relation used to bind the right dependencies of two tokens.

In our work, Stanford dependencies is used to identify pair of words that are adjacent to each other based on the index tagged next to it or words that tagged with grammatical relation of "*nn*" will be extracted to form a list of term. This is extremely useful to limit the generation of candidate terms from phrase structure trees.

Our hypotheses for detecting candidate terms are described as follows:

(i) Pair of words that are adjacent often partially contributes to entity recognition.
(ii) Pair of words that are tagged with "*nn*" grammatical relation often denotes important keywords for a domain.

2.2 WordNet

Wordnet[1] is the product of a research project at Princeton University which has attempted to model the lexical knowledge of a native speaker of English [14]. A derivationally related form (DRF) is one of the features available in WordNet being used to identify verb that associates with human action. In this work, each extracted verb (V) that formed S-V-O serves as a keyword for retrieving its corresponding senses' description in derivationally related forms. Each returned description will be examined sentence by sentence. In the presence of either one of the three key phrases of "*someone*", "*a person*" or "*one who*" in the sentence, the verb is considered to be associated with human action.

3 System Framework

The system framework for our proposed NER, focused solely in identifying fiction protagonist(s) is depicted in Fig. 1. The prototypical implementation of the automated NER (protagonist(s)) illustrated in Fig. 1 is explained as below:

The terms used in the framework are:

$term_{sd}$: Terms that are extracted from stanford dependencies based on two criteria; (1) words that are adjacent to each other and (2) words which are tagged with the "nn" grammatical relation
$term_{psp}$: Terms that are tagged with noun phrase (NP) in phrase dependency parse
$NE_{candicate}$: Candidate NE
$VERB_{per}$: Verb that associates with the human action

The first four steps of system framework in this work are generally similar to previous work done in ontology construction for fiction-based domain [12].

Input : Fiction web page
Step 1 : Document cleaning

[1] http://wordnet.princeton.edu/

Eight fairy tales are used to test on the proposed framework. Each fairy tale web page retrieved from the selected domain web pages is cleaned automatically using HTML Context Extractor[2] in order to get rid of non-text content (banner, audio, video, images). A pure text file (.txt) is produced at the end of the cleaning process.

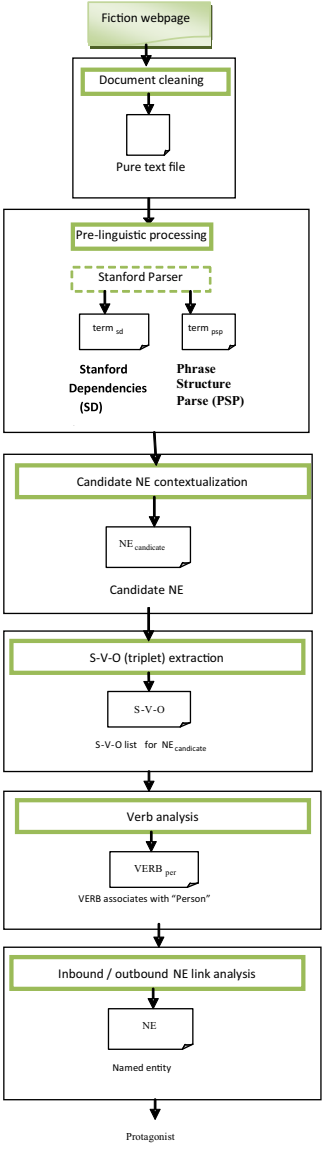

Fig. 1. System framework for NE (protagonists/main actor) recognition

[2] http://senews.sourceforge.net/KCE_README.html

Step 2 : Pre-linguistic processing

Two features available in Stanford parser which are phrase structure parse (PSP) and stanford dependencies (SD) will be used to shallowly examine the text content of the generated pure text file. Parse tree generated by PSP will be further manipulated by extracting phrases that tagged with "*NP*" to form list of term known as $term_{psp}$ whereas predicate argument structure produced by SD will be analyzed according to the two hypotheses mentioned in section 2.1.2 to form list of term known as $term_{sd}$.

Step 3 : Candidate NE Contextualization

NE often appears to be the subject and/or object of a sentence and it usually tagged as "*NP*" in a parse tree. However, not all "*NP*" correspond to NE. For instance, in this work, "*this*", and "*nothing*" are the $term_{psp}$ extracted from "*The Story of Snow White*". Therefore, instead of solely extracting all "*NP*" listed in PSP to form candidate NE. Nested or exact wording of $term_{sd}$ against $term_{psp}$ at sentence level is used to overcome the over generation of candidate NE as illustrated in equation (1) and describe elsewhere[12]. It implies parallelism in generating candidate NE.

$$term_{sd} \cap term_{psp} = NE_{candicate} \quad (1)$$

The overlap between $term_{sd}$ and $term_{psp}$ will result in utilizing $term_{psp}$ to form $NE_{candidate}$. This is due to NE may consist of more than two words while $term_{sd}$ always represents its grammatical relation in two words.

Step 4 : S-V-O (triplet) Extraction

A syntactic triplet structure of S-V-O denotes an event / action being take place between a subject (S) and an object (O). In this work, S implies 1st $NE_{candicate}$ and O marks 2nd $NE_{candicate}$. V is a Verb Phrase (VP) that exist between 1st $NE_{candicate}$ and 2nd $NE_{candicate}$ in a sentence basis. A sentence might have more than one S-V-O syntactic triplet structure. Extracted S can also be the O for another extracted triplet and vice versa.

Step 5 : Verb Analysis

"getDerivationallyRelatedForms" is one of the methods freely available in WordNet API (JAWS)[3] which is used to automatically examine against each verb that resides in the extracted triplets in the previous step. Each verb serves as a keyword for retrieving its corresponding senses' description in derivationally related forms. Key phrases of "*someone who*", "*one who*" or "*a person*" are the hints use to identify verb that associates with human activity. Therefore, each return description will be examined sentence by sentence to locate the

[3] http://lyle.smu.edu/~tspell/jaws/index.html

above mentioned hints. An integer value of 1 will be assigned to each S and O if the verb connected between them contains any of the three hints mentioned. At the end, each S and O might has a value of zero or more for its inbound link and outbound link that associated with human action related verb ($VERB_{per}$), as shown in Fig. 2.

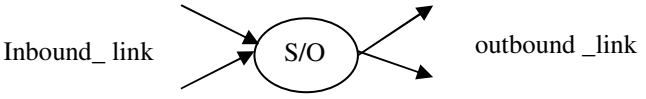

Fig. 2. Inbound and outbound links of S and O

Step 6 : Inbound/Outbound NE Link Analysis

Each inbound link and outbound link of $NE_{candicate}$ will be calculated for its proportional value as shown in equation (2) and (3). A filtering process of $NE_{candicate}$ is then done based on the calculated proportion. $NE_{candicate}$ which has value of 0, 1 and "#DIV/0!" for *in* and/or *out* will be discarded for further analysis in identifying protagonist. 0 means none of the inbound and/or outbound link is associated with human action, 1 denotes all inbound and/or outbound links are associated with human action and "#DIV/0!" shows division by zero, which denotes that no inbound or outbound link attached to $NE_{candicate}$. Later, normalization process of each filtered $NE_{candicate}$ will be performed based on three equations (4), (5) and (6). Equations (4), (5) and (6) imply the inbound link (Nin), outbound link (Nout) and frequency (Nfreq) respectively. Frequency signifies the sum of inbound and outbound links for each filtered $NE_{candicate}$. Finally, each calculated proportion indicates the weight carried by filtered $NE_{candicate}$. High proportion of these three measures may increase the likelihood that the filtered $NE_{candicate}$ is the protagonist of the investigated fairy tale.

$$\text{in} = \text{inbound\_link\_VERB}_{per} / \text{inbound\_link} \tag{2}$$

$$\text{out} = \text{outbound\_link\_VERB}_{per} / \text{outbound\_link} \tag{3}$$

$$\text{Nin}_i = \frac{\text{in}_i}{\sum_{i=1}^{N} \text{in}_i} \tag{4}$$

$$\text{Nout}_i = \frac{\text{out}_i}{\sum_{i=1}^{N} \text{out}_i} \tag{5}$$

$$\text{Nfreq}_i = \frac{\text{inbound\_link}_i + \text{outbound\_link}_i}{\sum_{i=1}^{N} (\text{inbound\_link}_i + \text{outbound\_link}_i)} \tag{6}$$

where 1 <= i <= N and N is the total number of filtered $NE_{candicate}$ in a fairy tale; Nin and Nout are the normalized values for inbound and outbound links respectively while Nfreq is the normalized value for frequency.

Output : Each proportion measured in the previous step is summed up to produce a unit measurement for each filtered $NE_{candicate}$ (equation (7)). Later, median of all weights is calculated to serve as a threshold value in identifying protagonist(s). Therefore, different fairy tales may have different threshold values. Finally, list of protagonist(s) is produced.

$$\text{weight} = \text{Nin} + \text{Nout} + \text{Nfreq} \quad (7)$$

4 Experiments and Discussions

4.1 Dataset

Eight fairy tales from http://www.kidsgen.com were used to test on our proposed framework and the word count for each fairy tales is presented in the second column of Table 1 [1]. The eight fairy tales were chosen as it reflects the aim of this work which is to identify protagonist(s) in diverse spectrums. Some of the protagonists have a character name while some are just the type of the animal/inserts.

4.2 Results and Discussion

The most challenging issue in NER, particularly in the fiction domain of fairy tales lies in its evaluation with gold standard as protagonist name might (1) slightly vary according to the version of the tales or (2) context sensitive to the local flavor. Therefore, a simple survey was conducted on the eight studied fairy tales on 6 primary school students. The third column in Table 1 shows protagonists for each fairy tales obtained from the survey. This result is used as a gold standard in our work to measure the performance of our approach and other NER tools. Three evaluation metrics, namely recall, precision and F-measure are used to evaluate the outcome of the extracted protagonist(s).

Table 1. Fairy tales word count and protagonists

Fairy Tale	Word Count	Protagonist
The Story of Snow White	1913	Snow White
Cinderella	1077	Cinderella
Beauty and the Beast	1357	Beauty, Beast
Rapunzel	1393	Rapunzel
Thumbelina	4348	Tiny
Ugly Duckling	841	Duckling
Sleeping Beauty	1317	Briar Rose
Ant and the Grasshopper	142	Ant, Grasshopper

Table 2 shows the protagonists extracted by our method using 2, 3 and 4 variables. The actual protagonist for each fairy tale appears in bold. 2-variable considered only proportion of inbound link (equation 4) and outbound link (equation 5) that attached to filtered $NE_{candicate}$. 3-variable is the method we have used in work as shown in Step 6 above while 4-variable is an extension of 3-variable with an additional proportion of

links (inbound and outbound link) that associates with $VERB_{per}$ against all links (inbound and outbound link) of each filtered $NE_{candicate}$. Normalization is performed on each calculated proportion. The number of extracted protagonist(s) is the same across 2, 3 and 4 variables. However, the protagonist that were extracted are slightly different between 2-variable and 3, 4-variable. 3-variable and 4-variable extract the same list of protagonists.

Table 2. Inbound/Outbound NE link analysis using different variables

Fairy Tale	2-variable	3-variable	4-variable
The Story of Snow White	**Snow White**	**Snow White**	**Snow White**
Cinderella	**Cinderella**	**Cinderella**	**Cinderella**
Beauty and the Beast	Merchant, Daughter, Horse	Merchant, Daughter, **Beauty**	Merchant, Daughter, **Beauty**
Rapunzel	Time, Enchantress	**Rapunzel**, Enchantress	**Rapunzel**, Enchantress
Thumbelina	Leaf, Mole, **Tiny**, Feather, Earth, Country, Heart	Leaf, Mole, **Tiny**, Feather, Earth, Flower, Bird	Leaf, Mole, **Tiny**, Feather, Earth, Flower, Bird
Ugly Duckling	null	null	Null
Sleeping Beauty	**Briar Rose**	**Briar Rose**	**Briar Rose**
Ant and the Grasshopper	**Ant**	**Ant**	**Ant**

As mentioned in step 6, it is insensible to naively accept protagonist which the $NE_{candicate}$ has only one inbound and/or outbound link, and the verb is associated with human action. Comparatively, $NE_{candicate}$ that has more than one inbound and/or outbound links will definitely have decreased verb probability associates with human action. In fact, protagonists are the main character(s) that should actively engage in story flow. From our experimental dataset, it is observed that there are at least two existences of inbound and/or outbound link for each corresponding $NE_{candicate}$. Finally, filtered $NE_{candicate}$ is produced after eliminating $NE_{candicate}$ that has the value of 0, 1 and "#DIV/0!" for its corresponding *in* and *out*. The above mentioned scenario reflects the approach applied in 2-variable, high proportional value of inbound and outbound link as approach taken in might not sufficient in identifying protagonist in fairy tales. The additional proportional value in 4-variable does not contribute to protagonist identification as it shows the same result as 3-variable. The engagement of filtered $NE_{candicate}$ in a story is shown explicitly by frequency (total link of inbound and outbound). High proportional value of each variable (3-variable) increases the probability of filtered $NE_{candicate}$ to be chosen as protagonist whereas high frequency with low proportional value of inbound and outbound link may prevent $NE_{candicate}$ to be protagonist.

Table 3 compares the results between our approach with three freely available tools on the internet, namely AlchemyAPI[4], General Architecture for Text Engineering

[4] http://www.alchemyapi.com/api/entity/

(GATE)[5] and Illinois named entity tagger[6]. Note that the result on "*Ugly Duckling*" is not included as none of the tools, including our approach is able to identify any protagonist in the story. AlchemyAPI employs hybrid approach in NER where statistical algorithms are combined with natural language processing technology to analyze and identify hundreds of entity types and "people" is one of its types. Contextual cues are used to disambiguate among entity types. For instance, information on a person's career and where they are located are some of the contextual cues used to disambiguate "people" entity type. GATE uses predefined gazetteer list (ANNIE) and rule based approach (JAPE) for finding entity types. Various machine learning techniques can also be imposed on GATE to increase the performance of the NER. However, in this paper, the default GATE NER was used. Illinois extracts NE using external knowledge (gazetteers) and machine learning paradigm. Portability, scalability and no training corpus are the main reasons of choosing these three tools for comparison.

Table 3. Comparison of performance metrics with other tools

		Recall	Precision	F-measure
Beauty and the beast	AlchemyAPI	0.0000	0.0000	0.0000
	GATE	0.0000	0.0000	0.0000
	Illinois	0.5000	1.0000	0.6667
	Our approach	0.5000	0.3333	0.4000
Cinderella	AlchemyAPI	1.0000	0.3333	0.5000
	GATE	0.0000	0.0000	0.0000
	Illinois	1.0000	0.5000	0.6667
	Our approach	1.0000	1.0000	1.0000
Rapunzel	AlchemyAPI	1.0000	1.0000	1.0000
	GATE	0.0000	0.0000	0.0000
	Illinois	0.0000	0.0000	0.0000
	Our approach	1.0000	1.0000	1.0000
Sleeping beauty	AlchemyAPI	1.0000	1.0000	1.0000
	GATE	0.0000	0.0000	0.0000
	Illinois	1.0000	1.0000	1.0000
	Our approach	1.0000	1.0000	1.0000
Snow white	AlchemyAPI	1.0000	1.0000	1.0000
	GATE	1.0000	0.3333	0.5000
	Illinois	0.0000	0.0000	0.0000
	Our approach	1.0000	1.0000	1.0000
Thumbelina	AlchemyAPI	0.0000	0.0000	0.0000
	GATE	0.0000	0.0000	0.0000
	Illinois	1.0000	1.0000	1.0000
	Our approach	1.0000	0.1429	0.2500
Ant and grasshopper	AlchemyAPI	0.0000	0.0000	0.0000
	GATE	0.0000	0.0000	0.0000
	Illinois	0.0000	0.0000	0.0000
	Our approach	0.5000	1.0000	0.6667

[5] http://gate.ac.uk/

[6] http://cogcomp.cs.illinois.edu/page/demo_view/8

Table 4. Average performance results for each tool

	Recall	Precision	F-measure
AlchemyAPI	50.00	42.67	43.75
GATE	12.50	4.17	6.25
Illinois	43.75	43.75	41.67
Our approach	75.00	68.45	66.46

In this work, protagonist can be generally divided into two categories, namely protagonist name (e.g., Snow White and Beauty) and protagonist entity, e.g., insects (ant, grasshopper), and animal (duckling). As can be seen in Table 3, most of the tools perform well on the protagonist name, except for GATE which is constrained by the limited number of name listed in gazetteer. The good performance in this aspect is due to protagonist name is very similar to human name. However, the three tools performed poorly and in fact none of the protagonist entity were identified. Comparatively, our approach is able to perform across the two mentioned categories. In addition to producing comparable performance results with the chosen three tools on the protagonist name, our approach outperforms the rest in identifying protagonist entity, which is insects, *ant* specifically in the fairy tale of "*Ant and grasshopper*". However, another protagonist entity of *grasshopper* is not identifiable as it has 0 value for *in* and that reduced the weight shown in equation (7) to be below the threshold. The same applies to the fairy tale of "*Ugly duckling*" that none of the tools is capable in identifying "*duckling*" as its protagonist because the word "*duckling*" does not seem to be human related name or appearing in the listed gazetteer. Our approach failed too, owing to the 0 value generated for *in* (equation 4). There is only one action imposed on "*duckling*" and the action is not related to human action. A protagonist should interact actively in story flow and contribute to inbound link ($VERB_{per}$, action being taken towards protagonist) and outbound link ($VERB_{per}$, action taken by protagonist). Therefore, both number of actions being taken and imposed on filtered $NE_{candicate}$, and its relevancy to human action give strong impact during protagonist identification. Lacking of either factor may hamper the effort of protagonist identification.

File size and activities/events affiliated with protagonist in fairy tale do impact the performance results of protagonist identification. This is due to the $Verb_{per}$ for inbound and outbound link will influence the proportion of $NE_{candicate}$'s inbound and outbound link. Small file size imposes limited activities or events affiliated with protagonist, which reduces the probability of inbound and outbound link that contain $VERB_{per}$. This can be seems very clearly for the fairy tales of "*Ant and grasshopper*" and "*Ugly duckling*" which have 142 and 841 word count respectively.

Recall, precision and F-measure are interdependent. High recall with low precision and vice versa might yield low F-measure, while high recall and high precision will definitely generate high F-measure. With the existence of one or two protagonists for a fairy tale always incur low precision if the number of identifiable filtered $NE_{candicate}$ is high and vice versa. Therefore, carefully taking care of each fairy tales nature is likely to improve the performance results of the protagonist identification. Table 4 summarizes the comparative study. Our approach yields better results compared to the other three tools.

5 Conclusion

This paper presents an algorithmic framework for protagonist identification in fiction domain. Comparatively, our proposed method is able to perform consistently. For future work, we intend to improve the protagonist identification by collaborating with VerbNet [15], which is the largest on-line verb lexicon in English that incorporates both semantic and syntactic about its content.

References

1. Chiticariu, L., Krishnamurthy, R., Li, Y.Y., Reiss, F., Vaithyanathan, S.: Domain Adaptation of Rule-Based Annotators for Named-Entity Recognition Tasks. In: Empirical Methods in Natural Language Processing, Massachusetts, pp. 1002 – 1012 (2010)
2. McCallum, A., Li, W.: Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In: 7th Conference on Natural Language Learning, pp. 188–191 (2003)
3. Klein, D., Smarr, J., Nguyen, H., Manning, C.D.: Named Entity Recognition with Character-Level Models. In: 7th Conference on Natural Language Learning, pp. 180–183 (2003)
4. Irmak, U., Kraft, R.: A Scalable Machine-Learning Approach for Semi-Structured Named Entity Recognition. In: 19th International World Wide Web Conference, North Carolina, pp. 461–470 (2010)
5. Le, H.T., Nguyen, T.H.: Name Entity Recognition using Inductive Logic Programming. In: Symposium on Information and Communication Technology, Vietnam, pp. 71–77 (2010)
6. Minkov, E., Wang, R.C., Cohen, W.W.: Extracting Personal Names from Email: Applying Named Entity Recognition to Informal Text. In: Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Vancouver, pp. 443–450 (2005)
7. Sekine, S., Nobata, C.: Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy. In: 4th International Conference on Language Resource and Evaluation (LREC), pp. 1977–1980 (2004)
8. Smarr, J., Manning, C.D.: Classifying Unknown Proper Noun Phrases without Context. Technical Report dbpubs/2002-46. Stanford University, Stanford, CA (2002)
9. Artiles, J., Amigo, E., Gonzalo, J.: The Role of Named Entities in Web People Search. In: Conference on Empirical Methods in Natural Language Processing, Singapore, pp. 534–542 (2009)
10. Fleischman, M., Hovy, E.: Fine Grained Classification of Named Entities. In: 19th International Conference on Computational Linguistics, pp. 1–7 (2002)
11. Elson, D.K., Dames, N., McKeown, K.R.: Extracting Social Networks from Literary Fiction. In: 48th Annual Meeting of the Association for Computational Linguistic, Uppsala, Sweden, pp. 138–147 (2010)
12. Goh, H.N., Kiu, C.C., Soon, L.K., Ranaivo, B.: Automatic Ontology Construction in Fiction-based Domain. International Journal of Software Engineering and Knowledge Engineering (2011) (in Press)
13. Klein, D., Manning, C.D.: Accurate Unlexicalized Parsing. In: 41st Meeting of the Association for Computational Lingusitic, pp. 423–430 (2003)
14. Stark, M.M., Riesenfeld, R.F.: WordNet: An Electronic Lexical Database. In: 11th Eurographics Workshop on Rendering (1998)
15. Verbnet, `http://verbs.colorado.edu/~mpalmer/projects/verbnet.html`