Identification of Character Adjectives from *Mahabharata*

Apurba Paul
JIS College of Engineering
Kalyani
West Bengal, India
apurba.saitech@gmail.com

Dipankar Das
Jadavpur University
188, Raja S.C. Mullick Road, Kolkata
West Bengal, India
dipankar.dipnil2005@gmail.com

Abstract

The present paper describes the identification of prominent characters and their adjectives from Indian mythological epic, *Mahabharata*, written in English texts. However, in contrast to the traditional approaches of named entity identification, the present system extracts hidden attributes associated with each of the characters (e.g., character adjectives). We observed distinct phrase level linguistic patterns that hint the presence of characters in different text spans. Such six patterns were used in order to extract the characters. On the other hand, a distinguishing set of novel features (e.g., multi-word expression, nodes and paths of parse tree, immediate ancestors etc.) was employed. Further, the correlation of the features is also measured in order to identify the important features. Finally, we applied various machine learning algorithms (e.g., Naive Bayes, KNN, Logistic Regression, Decision Tree, Random Forest etc.) along with deep learning to classify the patterns as characters or non-characters in order to achieve decent accuracy. Evaluation shows that phrase level linguistic patterns as well as the adopted features are highly active in capturing characters and their adjectives.

1 Introduction

The identification of characters from story texts has received a great significance in recent trends. Character (sometimes known as a fictional character) is a person or which may be other beings in a narrative work of art. In literature, characters guide readers through their stories, helping them to understand plots and ponder themes(Freeman,2016). According to Iosif (2014), characters in the stories can either be human or non-human entities, i.e., animals and non-living objects, exhibiting anthropomorphic traits. The interactions among characters can either be human-to-human or human-to-non-human interactions. Sometimes, it also shows the fact that a character may not be necessarily a speaker in context of stories. A character may appear in the story but may not have any quote associated with him/her. It means that it may not have any dialogue or monologue and hence, is not a speaker. Characters play the pivotal roles in order to comprehend the context and help the reader to understand the story in-depth. Thus, for identifying syntactic and semantic level narrative information from any story text, automatic character identification has always been of great significance. In general, we may find two types of characters such as protagonist or antagonist. A protagonist is the main character in any story and it can affect the decisions of main characters and propel the story forward(Duncan,2006). Similarly an antagonist also influences the story.

In a similar context, *The Mahabharata* is an ancient Indian epic where the main story revolves around two branches of a family, Pandavas and Kauravas, battles for the throne of Hastinapura. Interwoven into this, narrative and several smaller stories about people dead or living, and philosophical discourses are discussed in the epic. It is not that merely the names are the characters, often it is found that a noun phrase also refers to a character in an epic such as *The Mahabharata*. Thus, the extraction, identification and analysis of phrases related to a character is required in order to select important attributes with respect to that character.

So, the presence of adjectives and sometimes adverbs in the noun phrase are considered as the crucial attributes that help us to understand the character and its hidden qualities. For example, "*Yudhisthira*" is a character, and "*The Kuru King Yushisthira*" is also considered as a character containing its adjective, "*The Kuru King*" as shown in Example 1.

Proceedings of Recent Advances in Natural Language Processing, pages 569–576,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_074

Here, we understand that Yudhisthira is a character who is the king of Kuru dynasty.

Example 1: <*The Kuru King*$_{adj}$ Yudhisthira$_{character}$>

Similarly, in Example 2, "*Krishna*" is a character and "*the highly intelligent and high-souled Krishna*" should obviously be considered as a character. Here, Krishna has a quality of being "*highly intelligent and high-souled*" in the epic *The Mahabharata*.

Example 2: <*the highly intelligent*$_{adj}$ *and*$_{cc}$ *high-souled* $_{adj}$ *Krishna* $_{character}$>

Our goal of this research work is not only to identify characters from text but also to find out its attributive qualities, which we consider as character adjectives.

In this paper, we have formulated some rules which have been employed to extract a word, phrase or a group of words from the parsed sentences which is supposed to be a character. Then, the quality of these rules has been measured. A set of linguistic and statistical features was taken into consideration to identify different properties of the extracted word, phrase or group of words. Such textual units have been manually annotated as *Character* and *Not _a_Character* to prepare a complete tagged data set. Next, different classifiers have been applied on this data set to find out the precision, recall and f-measure; this has been followed by results and error analysis and observations.

In the rest of the paper, we have discussed related work and descriptions of the problem of character identification in *The Mahabharata*. Then, we have explained the data preparation steps followed by experiments, result and error analysis, and conclusion.

2 Related Work

A lot of works has been done on retrieving information from holy book Bible (English language), and Al-Quran (Arabic language). Mamade and Chaleira (2004) developed a system (DID) which was applied to children stories starts by classifying the utterances. The utterances belong to the narrator (indirect discourse) as well as belong to the characters taking part in the story (direct discourse). Afterwards, this DID system tries to associate each direct discourse utterance with the character(s) in the story. Goyal et al. (2010) proposed a system that exploits a variety of existing resources to identify affect states and applies "projection rules" to map the affect states onto the characters in a story. Calix et al. (2013) developed a methodology to detect sentient actors in the spoken stories. Valls-Vargas et al. (2013) proposed a method for automatically assigning narrative roles to characters in stories. Valls-Vargas et al. (2014) proposed a case-based approach to character identification in natural language text in the context of their Voz system. Valls-Vargas et al. (2015) also proposed a feedback-loop-based approach to identify the characters and their narrative roles where the output of later modules of the pipeline is fed back to earlier ones. In the context of keyword extraction, statistical approaches are often built for extracting general terms (Nees et al. 2010); the most basic measure is frequency. C/NC-value (Katerina et al. 2000), another statistical method is well known in the literature and combines statistical and linguistic information for the extraction of multi-word and nested terms.

3 Data Preparation

The English translation of the *Mahabharata* by *Kisari Mohan Ganguli* is the only complete one we can find in the public domain[1]. A total of 18 different chapters are present in the epic. The chapters are marked as *parva* (episode) e.g., *adi, sabha, vana, virata, udyog, bhishma, drona, karna, shalya, sauptika, stri, santi, anusasana, aswamedha, asramvasika, mausala, mahaprasthanika* and *svargarohanika*. It is observed that among the chapters, the 12th chapter, named as *santi parva* has the maximum number of sentences. This chapter contains 14929 uni-grams in a total of 23748 sentences. In contrast, the chapter 17 named as *mahaprasthanika parva* contains the minimum number of sentences and 888 uni-grams in a total of 188 sentences. As a whole, there are 120469 different sentences present in *Mahabharata*.

However, the average length of sentences in these chapters is significantly long (varies from 16 to 22 words). It was also found that *bhishma parva* (chapter 9) has 22 maximum number of average length sentences. Moreover, the characters occupy different floating slots within various text spans and the average number of character entities per sentence is 1.14 . Thus, the challenges lie in two spaces, one is to deal with sentences of varying length as well as to spot

[1] http://www.sacred-texts.com/hin/maha/index.htm

multiple characters appeared in different spans of a text. We have used Stanford CoreNLP[2] suite to tokenize the sentences and annotate them with Part-of-Speech (POS) tagger, syntactic parse tree etc. By analyzing the parsed sentences initially, it is observed that the NP which has VP as a right sibling is more tend to be the character in the text. Similar instances are observed when a NNP immediately follows a NP. In this regard, we have formulated a set of rules to extract the subtrees from the parsed sentence where NP holds the above mentioned properties and are considered as the entities. For an example,

R1: {NP<NNP $++ VP, NP<<-NNP}

where, {X<Y} means X immediately dominates Y in parse tree,

{X $++ Y} means X is a left sister of Y in parse tree, and {X<<-Y} means Y is the rightmost descendent of A in parse tree of a sentence.

Entity (e_1) = (NP (DT the) (ADJP (RB highly) (JJ intelligent) (CC and) (JJ high-souled)) (NNP Krishna))

*= [**the highly intelligent and high-souled Krishna**]*

*Entity (e_2) = (NP (NNP Krishna)) = [**Krishna**]*

The average Support ($\mathbf{S_{Avg}}$) and Confidence ($\mathbf{C_{Avg}}$) of each rule has been given in Table 1.

Rule #	Rules	S_{Avg}	C_{Avg}
R1	NP<NNP	55.45	64.34
R2	NP<NNP $++ (VP<VBD)	7.67	89.35
R3	NP<NNP $++ (VP<VBG)	1.00	83.11
R4	NP<NNP $++ (VP<VBN)	1.16	79.20
R5	NP<NNP $++ (VP<VBP)	0.48	64.10
R6	NP<NNP $++ (VP<VBZ)	0.99	75.21

Table 1: Average Support and Confidence of rules

A rule *R* can be assessed by its coverage and accuracy. Given a tuple X from a class labelled data set D , let N_{covers} be the total number of tuples covered by *R*; $N_{correct}$ be the total number of tuples correctly identified by *R*; and | D | be the total number of tuples in *D* . We can define the *Coverage* and *Accuracy* of *R* as follows.

$$\text{Coverage}(R)=\frac{N_{covers}}{|D|} \qquad \text{Accuracy}(R)=\frac{N_{correct}}{N_{cover}} \quad (1)$$

That is, a rule's coverage is the percentage of tuples that are covered by the rule. For rule's accuracy, we look at the tuples that it covers and see what percentage of them the rule can correctly identify . The observations of coverage and accuracy for each of the rules are described in Table 2. Here |D|=228810.

Rules #	N_{covers}	$N_{correct}$	Coverage %	Accuracy %
R1	**200522**	114053	**87.63**	56.87
R2	17471	15467	7.63	**88.52**
R3	2759	2230	1.20	80.82
R4	3205	2478	1.40	77.31
R5	2012	1137	0.87	56.51
R6	2841	2022	1.24	71.17

Table 2: Coverage and Accuracy of Rules

3.1 Quality Measures of Rules

Sometimes, we find that accuracy, on its own, is not a reliable estimate of judging quality of a rule. Even for a given class, we could have a rule that covers many tuples, but most of which belong to other classes. So, we need other measures for evaluating quality, which may integrate aspects of accuracy and coverage (Han and Kamber, 2009). Here, we look at three measures e.g., Entropy (R), FOIL_Gain and Likelihood Ratio statistics. The quality measures of each rule are given in Table 3.

Rule #	Entropy	FOIL_GAIN	Likelihood
R1	**0.29**	-2683.20	362.24
R2	-3.71	**2608.00**	**3043.77**
R3	-50.29	287.85	238.36
R4	-41.42	272.09	186.71
R5	-75.16	-29.94	2.66
R6	-48.40	149.30	66.4

Table 3: Quality Measures of Each Rule

In addition to such important rules, we have tried to extract more features for employing them in a ML framework.

In this paper, we have extracted two types of features, viz., linguistic features and statistical features for each of the entities. To the best of our knowledge, these features have not been yet explored in literature for character identification.

[2] https://stanfordnlp.github.io/CoreNLP

3.2 Linguistic Features

To extract the linguistic features for each of the rules (R_e), we have extracted the set of attributes explained below:

Current head node of extracted entity(C_h), The preterminal nodelist of C_h (P_l), the desired character adjective entity (C_{adj}), List of siblings of C_h (S_l), list of preterminal yields of all siblings of C_h (SP_l), path from C_h to two level up parent node ($Path_{2up}$), immediate ancestor node of C_h (AN_n), list of head nodes of siblings of immediate ancestor node from C_h (AN_l), list of preterminal yield nodes of siblings of immediate ancestor node from C_h (ANP_l).

Consider the rule R_e is *NP<NNP $++ (VP<VBD) and* a sentence S_1="*O ye ascetics,* ***the great Vyasa*** *hath composed one hundred and eighty-six sections in this Parva.*"

The corresponding parsed tree of the sentence S_1 is

S_{1p}=" (ROOT (S (NP-TMP (NP (NN O)) (NP (PRP ye) (NNS ascetics))) (, ,) **(NP (DT the) (JJ great) (NNP Vyasa))** (VP (VBP hath) (VP (VBN composed) (NP (NP (CD one) (CD hundred) (CC and) (CD eighty-six) (NNS sections)) (PP (IN in) (NP (DT this) (NN Parva)))))) (. .)))".

The linguistic features (C_h, P_l, C_{adj}, S_l, SP_l, $Path_{2up}$, AN_n) extracted from S_{1p} are shown in Figure 1 .

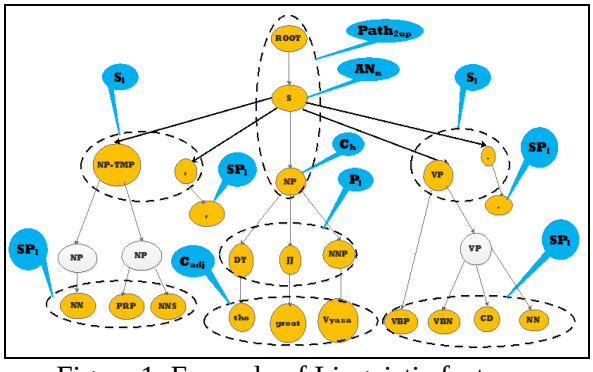

Figure 1: Example of Linguistic features

Again, consider another sentence S_2="***The mighty Jayatsena*** *the son of Jarasandha, the prince of the Magadhas, O king, hath been slain in battle by the high-souled son of Subhadra.*" and its corresponding parsed tree S_{2p} is:

S_{2p}="(ROOT (S (NP (NP **(NP (DT The) (JJ mighty) (NNP Jayatsena))** (NP (NP (DT the) (NN son)) (PP (IN of) (NP (NNP Jarasandha))))) (, ,) (NP (NP (DT the) (NN prince)) (PP (IN of) (NP (NP (DT the) (NNPS Magadhas)) (, ,) (NP (NNP O) (NN king))))) (, ,)) (VP (VBP hath) (VP (VBN been) (VP (VBN slain) (PP (IN in) (NP (NN battle))) (PP (IN by) (NP (NP (DT the) (JJ high-souled) (NN son)) (PP (IN of) (NP (NNP Subhadra)))))))) (. .)))"

The linguistic features (AN_l, ANP_l) extracted from S_{2p} are shown in Figure 2 below.

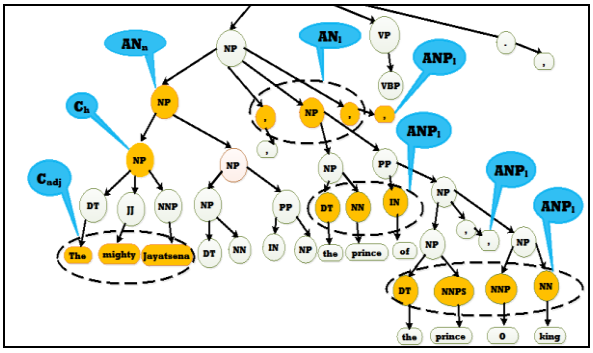

Figure2: Example of Linguistic features

Now, we have applied Stanford Universal Dependency parser[3] to each character adjectives entity (C_{adj}). The parser gives us the relation, governor and dependency of the words present in each NP entity as another set of features. They are relation name of C_{adj} (STR_n), Governor value of C_{adj} (STG_v), Governor tag of C_{adj} (STG_t), dependent value of C_{adj} (STD_v), dependent tag of C_{adj} (STD_t). In S_{2p} , we have C_{adj} ="***The mighty Jayatsena***". Its dependency relation, governor are explained below.

det(mighty/JJ , The/DT) ,
appos(mighty/JJ, Jayatsena/NNP)

There are two relation names of desired character adjective entity found, STR_n , are *det and appos* .The governor value of the desired character adjective entity, STG_v , is *mighty*. The Governor tag of desired character adjective entity, STG_t, is *JJ*. Then the Dependent value of desired character adjective entity, STD_v, are *The and Jayatsena* . After that the Dependent tag of desired character adjective, STD_t, are *DT and NNP*.

3.3 Statistical Features

To extract the statistical features, we have calculated the term frequency (TF) and term frequency-inverse document frequency (TF-IDF) of each character adjectives entity (C_{adj}) found in the dif-

[3] https://nlp.stanford.edu/software/stanford-dependencies.shtml

ferent chapters of *Mahabharata* corpus, considering each chapter as a separate document. The variance and standard deviation of TF-IDF are 0.01779 and 0.13341 respectively. In addition to that, we have calculated C-value and NC-value of each character adjectives entity (C_{adj}). In the list of entities, we have seen that there are single word entities as well as multiword term entities. The degree to which a linguistic unit is related to domain specific concepts is called *Termhood* (Katerina 2000). To find the *Termhood* of each entity, we have applied modified C-value function to all of them. C-value is a domain independent method which aims to improve the extraction of nested terms. The C-value assigns a *Termhood* to an entity, ranking it in the output list of each candidate character adjectives (C_{adj}).

$$\text{C\_value(a)}=\begin{cases}\log_2|a|.f(a) \text{ if a is not nested,}\\ \log_2|a|\left(f(a)-\frac{1}{P(T_a)}\sum_{b\in T_a} f(b)\right) \text{ otherwise}\end{cases} \quad (2)$$

Where,

a = candidate character adjectives entity (C_{adj}), b = longer entity, |a| = length of the entity (number of words), f(a) = frequency of a in the corpus, T_a = set of extracted candidate terms that contain a, $P(T_a)$ = number of candidate terms in T_a, f(b) = frequency of longer entity b in the corpus. After that, we have used NC-value method which incorporates context information into the C-value method. This method re-ranks the C-value list of each candidate character adjectives entity (C_{adj}). The NC-value measure is formally described as

$$\text{NC\_value}=0.8\text{C\_value(a)}+0.2\sum_{b\in C_a} f_a(b)w(b) \quad (3)$$

where, a= candidate character adjectives entity (C_{adj}), C_a= the set of distinct context words of a, f_a (b) = the frequency of b as a term context words of a, w(b)= weight of b as a term context word. As an example, if we consider C_{adj} =*"the Kuru king Yudhishthira"* then its C-value is 36.00 and NC-value is 46.56. The range of C-value varies from 0 to 7124.92 and the range of NC-value varies from 0.000022 to 193472.6998. In this way, we have collected all the linguistic and statistical features and arranged them in a set named as $Attr_{Total}$ for all the candidate character adjectives (C_{adj}).

$Attr_{Total}$ = { {R_e } ,{C_h, P_l , C_{adj}, S_l , SP_l, $Path_{2up}$, AN_n, AN_l , ANP_l},{STR_n, STG_v , STG_t, STD_v , STD_t}, {TF, TF-IDF}, {C-value, NC-value}}

From the above data preprocessing steps, we have manually tagged all the entities as *Character* and *Not_a_Character*. A total of 228810 objects extracted by the algorithm were manually annotated and for reference, two different independent domain experts were given the task of annotation to determine the *characters* and *non characters* in the dataset D according to their logic and perception. Secondly, they have identified the characters with their attributive qualities named as character adjectives. This identification task was done on the basis of few policies to identify a phrase as a character or character adjectives. Some of the important policies are as follows:

a)Every Name of a person followed by a verb is a Character.

e.g., <"*Yudhisthira*">

b) Each name of a person with its qualities which is often mentioned before or after the name is a Character.

e.g., < "*the wonderful warrior Drona* ">, <"*Arjuna the foremost* ">

c) Living, non living and celestial things which/who has done some action ,such as: speak, talk, walk or feel etc. and which has an active participation in the script is a *character*.

e.g., <"*the celestial Sakti*">,<"*the celestial Ganges*">

d) Each word which is related to some profession of a person (like: *sage*, *brahmana*) is a *character*.

e.g., "*The Asura architect*"

e) An animal which actively participates in the text is a *character*.

eg:<"*the celestial steed Uchchaihsrava*">

f) Any special weapon which is very powerful in case of destruction is termed as a Character because it has a particular identity, such as <"*the Sudarshana Chakra (the celestial disc)*">

eg: <"*the terrible weapon Narayana*">

To be very precise, every object of dataset D which has anthropomorphic trait is considered to be a character. The confusion matrices for identification of *character* and *Not_a_character* given by annotator-1 and annotator-2 are given in Table 4 and Table 5:

The **kappa measure** of agreement for identification of *Character* is **0.769** and for identification of *Not_a_Character* is **0.753**.

Character Identified in dataset D=137389		**Annotator 1**	
		Yes	**No**
Annotator 2	**Yes**	**135604**	360
	No	304	**1121**

Table 4: Confusion matrix of Characters/Character Adjectives by Annotator 1 and 2

Not _a_Character Identified in dataset D=91421		**Annotator 1**	
		Yes	**No**
Annotator 2	**Yes**	**88364**	632
	No	552	**1873**

Table 5: Confusion matrix of Not_a_Character by Annotator 1 and 2

4 Experiments

We have conducted our experiments on the data set, D that contains 15 linguistic features and 4 statistical features with 228810 entities. The data set seems to a two class problem, where each entity is manually tagged as *Character* or *Not_a_Character*. In order to satisfy the requirements of different classifiers, data preprocessing was conducted to convert textual information into numeric values. We have conducted feature ablation studies in two stages, one at the individual feature level using different attribute selection measures and another at subset level using *Forward Selection* and *Backward Elimination* schemes. Finally, we have applied different classification algorithms available under RapidMiner Studio tool[4] on our data set along with important attributes.

4.1 Attribute Selection Measure

Here, we have applied some popular attribute selection measures like information gain (I_g), gain ratio (G_r), gini index (G_i), Chi Squared Statistic (Chi) to our data set. The results are given below in Table 6. The attribute with highest **Information gain (I_g)** is chosen and the top three attributes with highest Information gain are **{C_{adj} ,P_l ,SP_l}**. The attribute with the maximum **Gain Ratio (G_r)** is selected as the splitting attribute. The top three attributes with highest Gain Ratio are **{NC-value, TF-IDF, C-value}**. On the other hand, **Gini Index (G_i)** is a measure of impurity of any data set. The higher the weight of an attribute, it is considered to be more relevant. The top three relevant attributes can be found from our data set are **{C_{adj} , P_l , STD_v}**. The **Chi-Square** statistic is a nonparametric statistical technique used to determine if a distribution of observed frequencies differs from the theoretical expected frequencies. The higher the Chi-Square value of an attribute, the more relevant it is considered . From our data set, the top three attributes selected using Chi-Square are **{ANP_l ,C_{adj} ,SP_l}**.

Attribute	**I_g**	**G_r**	**G_i**	**Chi**
R_e	0.028	0.038	0.016	7804.304
C_h	0	0	0	0
P_l	0.623	0.126	0.339	161885.5
C_{adj}	**0.924**	0.082	**0.461**	303530.6
S_l	0.261	0.045	0.152	73003.12
SP_l	0.575	0.058	0.297	269168.3
$Path_{2up}$	0.332	0.039	0.183	114295.3
AN_n	0.063	0.035	0.04	18843.35
AN_l	0.18	0.03	0.107	51561.98
ANP_l	0.435	0.05	0.226	**337090.9**
STR_n	0.425	0.142	0.25	119360
STG_v	0.503	0.071	0.278	209395.5
STG_t	0.208	0.116	0.128	61067.99
STD_v	0.541	0.097	0.304	145828.3
STD_t	0.407	0.146	0.24	114517.4
TF	0.121	0.287	0.072	33953.84
TF-IDF	0.189	0.335	0.115	40422.39
C-value	0.149	0.312	0.082	34288
NC-value	0.198	**0.348**	0.119	34571.75

Table 6: Attribute Selection measures

4.2 Feature Subset Selection

We have used two different schemes, e.g. *Forward Selection* and *Backward Elimination* available in Rapid Miner to find out different groups of relevant attributes or features,. Using the *Forward Selection* scheme, we have obtained a new set of attributes FS_F which is a subset of $Attr_{Total}$.

FS_F = { R_e , P_l ,C_{adj}, STG_t STD_v , STD_t, TF, TF-IDF } ;where $FS_F \subset Attr_{Total}$

Next, using the Backward Elimination scheme, we have acquired a new set of attributes BE_F which is also a subset of $Attr_{Total}$.

BE_F= { R_e ,C_h ,P_l , C_{adj}, S_l , SP_l, $Path_{2up}$, AN_n, AN_l , ANP_l, STG_t, STD_v , STD_t, TF, TF-IDF }

where $BE_F \subset Attr_{Total}$

[4] https://rapidminer.com/products/studio/

In both the schemes, we received a list of attributes as an end product. Then, we have prepared two different data sets D_{FS} and D_{BE} with the relevant attributes.

4.3 Classification Task

First, we have converted our data sets (D, D_{FS}, D_{BE}) into their compatible formats that are acceptable to the classifiers under Rapid Miner tool (e.g. *Deep Learning Classifier*, *KNN Classifier*, *Logistic Regression Classifier*, *NaiveBayes'*, *Decision Tree and Random Forest Classifier*). We have divided the datasets in 7:3 ratio for training and testing. Then, we have applied these classifiers on our data sets to find the accuracy along with different statistical measures.

5 Result Analysis

The detail observation of Precision , Recall and F-Measure for each classifier applied on the data set D, D_{FS} and D_{BE} are given in the Figure 3 and Table 7.

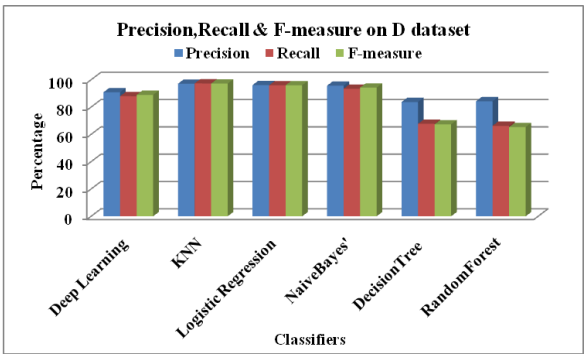

Figure 3: Precision, Recall and F-measure of classifiers on D dataset

It has been observed that KNN classifier obtained the highest Precision, Recall and F-measure on D dataset, whereas RandomForest has the worst Recall and F-measure on D dataset.

	D_{FS} dataset			D_{BE} dataset		
	P	**R**	**F**	**P**	**R**	**F**
Deep Learning	88.07	87.03	87.54	89.21	87.23	88.20
KNN	**92.19**	**92.2**	**92.19**	**95.66**	**95.66**	**95.66**
Logistic Regression	85.49	84.43	84.95	85.43	84.43	84.92
NaiveBayes'	**83**	75.81	79.24	81.59	74.1	77.66
DecisionTree	91.94	90.58	91.25	91.95	90.59	91.26
RandomForest	90.29	88.05	89.15	90.99	89.17	90.07
P=Precision; R=Recall; F=F-measure						

Table 7: Precision, Recall and F-measure on D_{FS} and D_{BE} datasets

Similarly, we have observed the Precision, Recall and F-Measure on D_{FS} and D_{BE} datsets as shown in Table7. Here, we found that KNN classifier achieved the best results among the other classifers for both the datasets. But, NaiveBayes' has the worst results for both the datasets. The confusion matrices for each classifier applied on the same data set D, D_{FS} and D_{BE} are observed. Here, N and C are *Not_a_Character* and *Character* respectively.

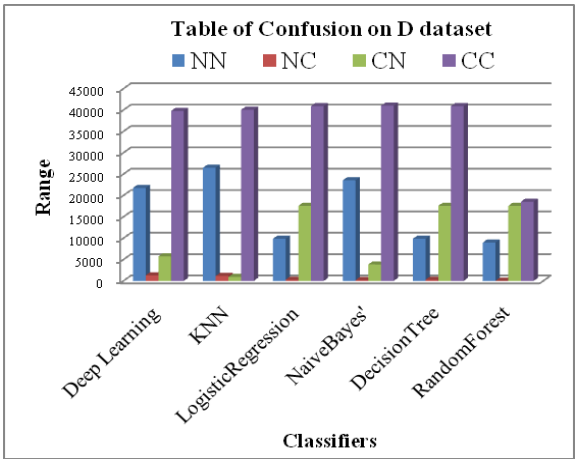

Figure 4: Confusion Table on dataset D

We observed from Figure 4 that NaiveBayes' classifier has maximum number of CC (*Character*, *Character*) and KNN classifier has maximum number of NN (*Not_a_Character*, *Not_a_Character*) confusions on dataset D.

	D_{FS} dataset				D_{BE} dataset			
	NN	**NC**	**CN**	**CC**	**NN**	**NC**	**CN**	**CC**
Deep Learning	7418	200	2283	9759	7687	515	2014	9444
KNN	**9298**	450	403	9509	8985	819	716	**9140**
Logistic Regression	7417	757	2284	9202	7393	732	2308	9227
NaiveBayes'	4856	185	4845	9774	5121	117	4580	9842
DecisionTree	7963	91	1738	9868	7964	93	1737	9866
Random Forest	7677	79	2024	**9880**	7444	63	2257	**9896**
N=Not_a_Character; C=Character								

Table 8: Confusion Table on D_{FS} and D_{BE} datasets

From the Table 8 we can observe that KNN classifier has maximum number of NN (*Not_a_Character*, *Not_a_Character*) and RandomForest classifier has maximum number of CC (*Character*, *Character*) in both the datasets D_{FS} and D_{BE} respectively.

6 Error Analysis and Observations

	Classification error(%)		
	D dataset	**D_{FS} dataset**	**D_{BE} dataset**
Deep Learning	10.29	12.63	12.86
KNN	**2.67**	4.34	7.81
Logistic Regression	3.83	15.47	15.46
NaiveBayes'	5.35	**25.58**	**23.89**
DecisionTree	25.94	9.3	9.31
RandomForest	27.08	10.7	11.8

Table 9: Error Rate of D, D_{FS} and D_{BE} datasets

It is observed from Table 9 that KNN classifier has the lowest error rate on all the datasets and it implies that KNN has the best performance over other five classifiers. On the other hand, we observed that random classifiers have the worst performances on dataset D and NaiveBayes' has the highest error rate on dataset D_{FS} and D_{BE}.

7 Conclusion

In this paper, we have presented a novel approach to identify *Characters* and *Character Adjectives* from unannotated Indian mythological epic called *Mahabharata* depending on some phrase level rules. Then, we have applied a couple of machine learning algorithms to classify whether an extracted object using the predefined rule is a character/character adjective or not.The experimental results showed that our approach delivers the best results when we have applied KNN classifier followed by Logistic Regresion, NaiveBayes and Deep Learning classifiers. We have also shown that a set of features are very important in classification using feature subset selection schemes. As part of the future work, we have planned to create a larger set of phrase level rules for better evaluation of charcaters and character adjectives.

Acknowledgement

The present work is supported by "Young Faculty Research Fellows of the Visvesvaraya PhD Scheme" of MeitY, Govt. of India.

References

Amit Goyal, Ellen Riloff; and Hal D III.2010. *Automatically producing plot unit representations for narrative text.* In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp.77–86. Association for Computational Linguistics.

Ellias Iosif, Taniya Mishra, *From Speaker Identification to Affective Analysis: A Multi-Step System for Analyzing Children Stories*.2014. Proceedings of the 3rd Workshop on Computational Linguistics for Literature(CLfL)@EACL 2014, pp. 40-49.

Jiwaei Han and Micheline Kamber.2009.*Data Mining Concepts and Techniques.*

Josep Valls-Vargas J, Santiago Ontanon ,and Jichen Zhu . 2013.*Toward character role assignment for natural language stories.* Proceedings of the Ninth Artificial Intelligence and Interactive Digital Entertainment Conference.

Josep Valls-Vargas J, Santiago Ontanon and Jichen Zhu . 2014. *Toward automatic character identification in unannotated narrative text.* In Proceedings of the Seventh Workshop in Intelligent Narrative Technologies.

Josep Valls-Vargas, Jichen Zhu, Santiago Ontanon. 2015.*Narrative hermeneutic circle: Improving character role identification from natural language text via feedback loops*. Proceedings of the 24th International Conference on Artificial Intelligence. AAAI Press.

Katerina Frantzi, Sophia Ananiadou and Hideki Mima. 2000. *Automatic Recognition of Multi-Word Terms: the C-value/NC-value Method.* International Journal of Digital Libraries, 3(2), pp.117-132

Matthew Freeman. 2016. *Historicising Transmedia Storytelling: Early Twentieth-Century Transmedia Story Worlds.* Routledge. ISBN 1315439506.

Nees J V Eck , Ludo Waltman L, Ed C M Noyons Ed, and Reindert K Buter. 2010. *Automatic term identification for bibliometric mapping*. SpringerLink, Scientometrics,Volume 82, Number 3.

Numo Mamede and Pedro Chaleira .2004.*Character identification in children stories.* Advances in natural language processing. Springer Berlin Heidelberg, 2004, pp.82-90.

Ricardo A. Calix, Leili Javadpour, Mehdi Khazaeli, and Gerald M. Knapp. 2013. *Automatic Detection of Nominal Entities in Speech for Enriched Content Search.* The Twenty-Sixth International FLAIRS Conference, pp. 190–195

Stephen V Duncan. 2006. *A Guide to Screenwriting Success: Writing for Film and Television.* Rowman & Littlefield. ISBN 9780742553019