A Deep Dive into Identification of Characters from Mahabharata

Apurba Paul and Dipankar Das

Dept. of Computer Science and Engineering,
Jadavpur University,Kolkata,West Bengal,India

Abstract

The present paper describes the identification of story Characters from Indian Mythological text "The Mahabharata". It is observed that these Characters can be found at word level and phrase level in a sentence with some distinct patterns. In order to find the Characters from the text, two sets of features are considered at both levels. Using a semi-supervised learning approach we have prepared the training data sets. Later on, we have employed Chi-squared statistic to find the important features, which is followed by the associativity analysis of those selected features. After that, we developed training models using NeuralNet and KNN classifiers for both word and phrase levels and tested the models. Our observation shows that NeuralNet performs better than KNN with 88% and 76% accuracy at word and phrase level respectively.Next, we have analyzed different error measures followed by visualization of co-occurred story Characters of most frequent Characters.

1 Introduction

Characters has a significant role and takes part in several activities throughout any stories. They may or may not be lovable, respectable, honorable, graceful, disgraceful, cruel, selfish but the readers do need to understand them and why they act the way they do in the texts. A character, being a protagonist, is commonly on the good side while the antagonist is the one he/she fights or has conflicts within the story. In the stories, Characters may have dialogues, actions which influence the plot of the texts, emotions etc. They respond to events and other characters through what they say or don't say, what they do and don't do, what they think, and what they feel. Character's thoughts in response to the actions or words of others are obviously a key to that Character's personality. Like thoughts, Characters emotions can instantly reveal a Character's personality and what he/she finds important. If we dive deep in the story we can extract the actions, thoughts, emotions and overall personality of a Character easily. Since the Characters are playing the major role in any story, we can consider automatic identification of Characters from stories is one of the primary task .

In the similar context, we can find several Characters in the Indian epic "The Mahabharata". Here the Characters may be protagonist or antagonist. So the extraction and identification of Characters are very important. In this text we can find that a Character may appear in word level(NNP) or it may appear in a phrase level(NP<<NNP). As an example, "Yudhisthira"(NNP) is a Character at word level and simultaneously "The Kuru king Yudhishthira" (NP<<NNP) is also a Character at phrase level. But it is also seen that only NNP or NP<<NNP are not sufficient rule to identify a Character in the texts. So the identification of Characters at word and phrase level is the main research issue addressed over here.

In this paper, we have employed two different approaches to address the presence of a Character in Mahabharata. We have identified at word level a set of 97 features and at phrase level a set of 51 features. With the help of these features we have developed our data sets and later on devised two different training models using semi supervised approach. Next, we identified the set of important features and their associativity among them. After that we have tested our model and observed the precision, recall, f-measure, kappa and errors.

In the rest of the paper, we have discussed related work and the data preparation steps followed by

Proc. of ICON-2017, Kolkata, India. December 2017 ©*2017 NLPAI*, pages 447–455

experiments, result and error analysis, visualization of co-occurred Characters and conclusion.

2 Related Work

There are a few works done on Character Identification from texts. Paul and Das (2017) proposed a rule based system by which they can extract the Character Adjectives from the Indian mythological text Mahabharata. 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. 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. (2013) proposed a method for automatically assigning narrative roles to characters in stories. Calix et al. (2013) developed a methodology to detect sentient actors in the spoken stories. Goyal et al. (2010) proposed a system that exploits a variety of existing resources to identify affect states and applies to map the affect states onto the characters in a story. Mamede 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. In the context of keyword extraction, statistical approaches are often built for extracting general terms (Van Eck et al., 2010).

3 Data Preparation

In this paper we consider Mahabharata as a case study from where we choose aswamedha, asramvasika, mausala, mahaprasthanika and svargarohanika parva(or Chapter) as our sample space. We can observe that there exists a lot of Characters which plays a significant role in these texts. At first we annotate these Characters manually and made a list of Characters out of it . Then to understand the positions and occurrences of each Characters we investigate each sentences in the texts with the help of Stanford CoreNLP suite. We tokenized each sentences, annotate them with POS tagger and generate syntactic parse tree by the suite. After a detail observation of each sentence in each text we developed a notion that Characters can be found in word level and phrase level as well. We also observed that in most of the cases at word level, a word is a Characters when its POS tag is NNP. Similarly at phrase level, a phrase is a Character when the root of the phrase is NP and one of its descendant is NNP. The examples are given below.

At word Level:
(NNP Narayana)=[Narayana]$_{\text{Character}}$
At phrase Level:
(NP (DT the) (JJ holy) (NNP Rishi) (NNP Vyasa)) = [The holy Rishi Vyasa]$_{\text{Character}}$

3.1 Feature set Generation

The above observation helps us to extract different features at word level and phrase level. The list of features at both the levels with appropriate examples are explained in the next sub section.

3.1.1 Word Level Features

For each NNP present in a sentence at word level we have considered 97 different features. They are displayed in Table 1:

Word Level Features(WL_F)		
Sl(W)	**Name**	**Freq.**
1	Extracted NNP word(Cw)	4152
2	NNP-tag	4152
3	Length of Cw	4152
4	Starting Index of Cw	4152
5	Ending Index of Cw	4152
6	Previous word of Cw	3583
7	Previous word tag of Cw	2584
8	Next word of Cw	4152
9	Next word tag of Cw	4152
10	Porter Stemmer word of Cw	4152
11	Is porter Stemmed word same with Cw?	4152
12	Snowball Stemmer word of Cw	4152
13	Is snowball stemmed word same with Cw?	4152
Immediate Pre and Post . . . Features of Cw		
14-17	verb word and tag	2645,3158
18-21	adverb word and tag	1218,1381
22-25	preposition word and tag	2590,3042
26-29	noun word and tag	2741,3412

Continued on next page

Continued from previous page		
Sl(W)	**Name**	**Freq.**
30-33	NNP word and tag	2199,2229
34-37	adjective word and tag	1445,2089
38-41	C. Conjunc. word and tag	1052,1611
42-45	determiner word and tag	2608,2554
46-49	existential word and tag	0092,0043
50-53	interjection word and tag	0001,0001
54-57	TO word and tag	0541,0959
58-61	Cardinal Number and tag	0255,0263
62-65	pronoun word and tag	1162,1773
66-69	Wh word and tag	0625,0771
Immediate Pre and Post ... Distance from Cw		
70,71	verb distance	2645,3158
72,73	adverb distance	1218,1381
74,75	preposition distance	2590,3042
76,77	noun distance	2741,3412
78,79	NNP distance	2199,2229
80,81	adjective distance	1445,2089
82,83	C Conjunction distance	1052,1611
84,85	determiner distance	2608,2554
86,87	existential distance	0092,0043
88,89	interjection distance	0001,0001
90,91	TO distance	0541,0959
92,93	Cardinal Number distance	0255,0263
94,95	pronoun distance	1162,1773
96,97	Wh distance	0625,0771
		Concluded

Table 1: List of Word Level Features(WL_F)

In the Table 1, mainly we have categorized the set of features in three different sub categories.The features from W1 to W13 are sub categorized as general features of a context word(Cw), from W14 to W69, the features are sub categorized as Immediate pre and post word and tag of Cw and from W70 to W97, the features are responsible for counting the word distance from the context word Cw as immediate pre and post word distance, along with their frequencies.Here frequency reveals the number of occurrences of a distinct feature in our sample space. As an example, consider a feature set W14-17(verb word and tag). It contains four different types of features.The W14 is immediate pre verb word which is situated in the left of Cw and W15 is its POS Tag with frequency 2645. Next, W16 is immediate post verb word situated in the right side of Cw in a sentence and W17 identifies its POS Tag with frequency 3158. Again as an example consider W70,71(verb distance). Here W70 calculates the word distance of verb situated in the left of Cw as immediate pre verb distance.The frequency of this feature is 2645. Likewise, the W71 finds the word distance of verb situated in the right of Cw as immediate post verb with frequency 3158.

Consider a sentence $\mathbf{S_1}$= *"Having bowed down unto Narayana, and to Nara, the foremost of men, as also to the goddess Sarasvati, should the word Jaya be uttered."*

In the above sentence our context word(Cw) is $\mathbf{Narayana_{Character}}$. Some of the features extracted from the sentence $\mathbf{S_1}$ with respect to $\mathbf{Narayana_{Character}}$ are explained in Figure 1.

3.1.2 Phrase Level Features

At phrase level, we have considered 51 different features displayed in Table 2 for each NP<<NNP pattern present in the sentences.

Phrase Level Features(PL_F)		
Sl(P)	**Name**	**Freq.**
1	Current head Node of the phrase(Ch)	2991
2	The pre terminal yield Nodes of Ch	2991
3	Leaves of the Ch(Cw)	2991
4	Path from Ch to ancestor Node	2991
5	has ADJP as siblings of Ch?	0018
6	has ADJP as siblings of Ch?	0128
7	has CONJP as siblings of Ch?	0006
8	has FRAG as siblings of Ch?	0001
9	has INTJ as siblings of Ch?	0001
10	has LST as siblings of Ch?	0001
11	has NAC as siblings of Ch?	0001
12	has NP as siblings of Ch?	0790
13	has NX as siblings of Ch?	0001
14	has PP as siblings of Ch?	0271
15	has PRN as siblings of Ch?	0016
16	has PRT as siblings of Ch?	0001
17	has QP as siblings of Ch?	0001
18	has RRC as siblings of Ch?	0003
19	has UCP as siblings of Ch?	0003
20	has VP as siblings of Ch?	0619
21	has WHADJP as siblings of Ch?	0001
		Continued...

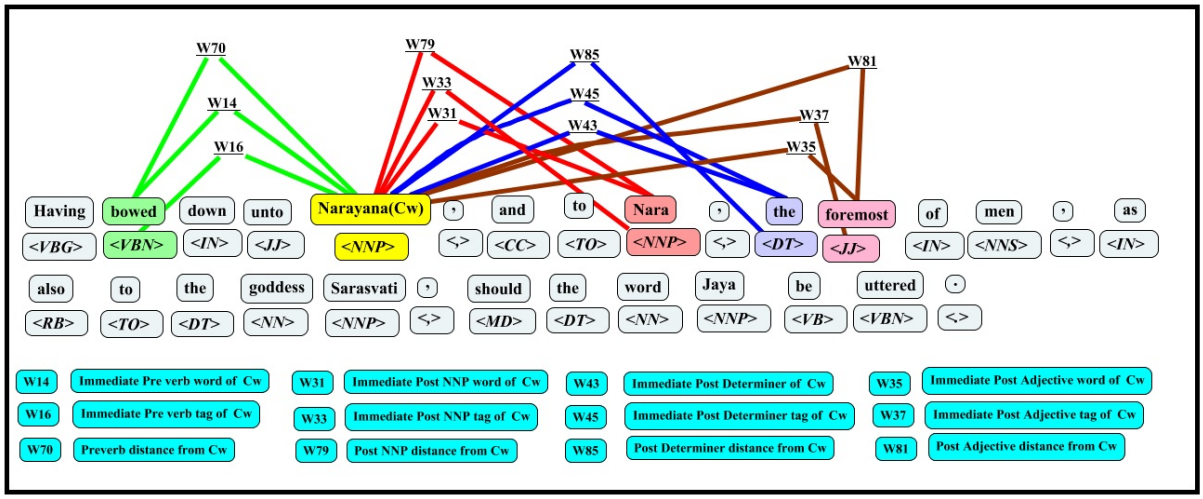

Figure 1: Example of Word Level Features

Continued...		
Sl(P)	**Name**	**Freq.**
22	has WHAVP as siblings of Ch?	0001
23	has WHNP as siblings of Ch?	0001
24	has WHPP as siblings of Ch?	0001
25	has X as siblings of Ch?	0004
26	has COMMA as siblings of Ch?	0788
27	has STOP as siblings of Ch?	0313
28	Ancestor Node of Ch(AnCh)	2991
29	has ADJP as siblings of AnCh?	0008
30	has ADJP as siblings of AnCh?	0088
31	has CONJP as siblings of AnCh?	0028
32	has FRAG as siblings of AnCh?	0001
33	has INTJ as siblings of AnCh?	0001
34	has LST as siblings of AnCh?	0001
35	has NAC as siblings of AnCh?	0001
36	has NP as siblings of AnCh?	1239
37	has NX as siblings of AnCh?	0001
38	has PP as siblings of AnCh?	0210
39	has PRN as siblings of AnCh?	0016
40	has PRT as siblings of AnCh?	0018
41	has QP as siblings of AnCh?	0001
42	has RRC as siblings of AnCh?	0001
		Continued...

Continued...		
Sl(P)	**Name**	**Freq.**
43	has UCP as siblings of AnCh?	0001
44	has VP as siblings of AnCh?	0470
45	has WHADJP as siblings of AnCh?	0001
46	has WHAVP as siblings of AnCh?	0001
47	has WHNP as siblings of AnCh?	0025
48	has WHPP as siblings of AnCh?	0001
49	has X as siblings of AnCh?	0001
50	has COMMA as siblings of AnCh?	0706
51	has STOP as siblings of AnCh?	0446
		Concluded

Table 2: List of Phrase Level Features(PL_F)

In the Table 2 it can be observed that there are mainly two different subcategories of features. All the features from P1 to P27 are related to the phrase(Cw) which is assumed to be a Character, and rest of the features are related to the two level up ancestor(parent of a parent of Current head node,Anch), along with their frequencies. Here frequency identifies the number of occurrences of a particular feature in the sample space. As an example, P1 contains the Current head Node of the phrase(Ch) with frequency 2991 and P36 finds the existence of any NP as a sibling of Ancestor Node of Ch(AnCh). The frequency of P36 is 1239.

Again Consider a sentence $\mathbf{S_2}$= *"The king, in honour of Hari and naming him repeatedly, fed* ***the Island-born Vyasa****, and Narada, and Markandeya possessed of wealth of penances, and Yajnavalkya of Bharadwaja's race, with many delicious viands."*

The important part of the parse tree of the above sentence $\mathbf{S_2}$ is ,

$\mathbf{S_{2parsed}}$= (VP (VBN fed) (NP (NP (DT the) (JJ Island-born) (NNP Vyasa)) (, ,) (CC and) (NP (NNP Narada))))

In the above sentence our target phrase is **the Island-born Vyasa$_{\mathbf{Character}}$**. Figure 2 explains the features P1,P2,P3 and P28 in details.

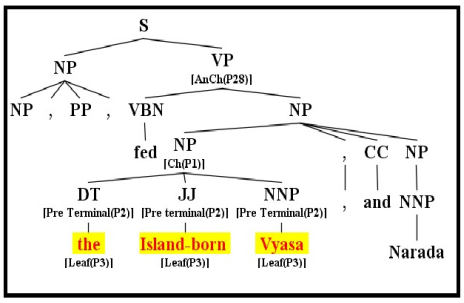

Figure 2: Example of Phrase Level Features P1,P2,P3,P28

3.2 Training & Test set Preparation

To prepare the training sets for both word level and phrase level we consider semi supervised learning approach. At first, we have extracted all the features of each NNP present in mahaprasthanika parva at word level and compare each NNP with manually tagged list of Characters. The NNP's which are found in the list are annotated as Character and in case of unavailability they are termed as Not_Character. In this way we have prepared a data set, $WD_{training}$. Next, we have extracted all the features of each NNP present in svargarohanika parva and prepared a dataset called WD_{test} . After that we developed a learning model trained on $WD_{training}$ data set using KNN-classifier and test the model using WD_{test} . Then we calculate the accuracy, precision, recall and f-measure of WD_{test}. Later on all the NNP's in WD_{test} dataset are annotated properly and update the $WD_{training}$ dataset by appending newly an-notated WD_{test} dataset. This process is repeated for all other chapters in our sample space. Finally we got an updated dataset $WD_{training}$ which is considered as a training set containing word level features for our system. The results are discussed in Table 3.

Word Level ($WD_{Training}$)			
Parva	**P**	**R**	**F**
svargarohanika	0.43	0.44	0.43
mausala	0.62	0.64	0.60
asramvasika	0.63	0.63	0.63
aswamedha	0.71	0.69	0.68
P=Precision; R=Recall; F=F-measure			

Table 3: Precision, Recall, F-measure at Word Level

Similarly, at phrase level we have extracted all the features of each NP<<NNP present in the mahaprasthanika parva and prepared a data set named as $PD_{training}$ and trained a model with KNN Classifier. Next, we have extracted all the features of each NP<<NNP present in svargarohanika parva and prepared a dataset called PD_{test} which is applied on the trained model like word level process. Here, we calculate precision, recall and f-measure of PD_{test} . This process is iterated for other chapters and finally we got updated $PD_{training}$ as a training set of phrase level. The results are observed in Table 4.

Phrase Level ($PD_{Training}$)			
Parva	**P**	**R**	**F**
svargarohanika	0.52	0.56	0.50
mausala	0.67	0.65	0.63
asramvasika	0.60	0.62	0.60
aswamedha	0.58	0.52	0.44
P=Precision; R=Recall; F=F-measure			

Table 4: Precision, Recall, F-measure on Phrase Level

At last we choose virata parva as a test case, annotate all the Characters present in the text and made a list out of it. Next, we prepared the data sets for word and phrase level, WT_{Test} and PT_{Test}respectively. Then we have mapped all the NNP, NP<<NNP present in the virata parva with Character and Not_Character which we will refer to in the Result Analysis section.

4 Experiments

To find the important features in $WD_{Training}$ and $PD_{Training}$ datasets we calculated the relevance of the features by computing the Chi squared statistic with respect to the Class level feature

using Rapid Miner tool[1]. The higher the weight of a feature, the more relevant it is considered. The value of the Chi Squared statistic is given by

$$X^2 = \sum \frac{(O-E)^2}{E} \quad (1)$$

where, X^2 is the chi-square statistic, O is the observed frequency and E is the expected frequency. Using this measure we got 46 important feature at word level and 9 at phrase level. The list of relevant features derived from the measure at word level(D_w) is given in Table 5.

Word Level Features(D_w)	
Sl	**Name**
W1	Extracted NNP-word(Cw)
W4	Starting Index of Cw
W5	Ending Index of Cw
W6	Previous word of Cw
W7	Previous word tag of Cw
W8	Next word of Cw
W9	Next word tag of Cw
W10	Porter Stemmer word of Cw
W11	Is porter Stemmed word same with Cw?
W12	Snowball Stemmer word of Cw
W13	Is snowball stemmed word same with Cw?
Immediate Pre and Post ... Features of Cw	
W14	Pre verb word
W15	Pre verb tag
W16	Post verb word
W20	Post adverb word
W22	Pre preposition word
W23	Pre preposition tag
W26	Pre noun word
W27	Pre noun tag
W28	Post noun word
W29	Post noun tag
W30	Pre NNP word
W31	Pre NNP tag
W32	Post NNP word
W35	Pre adjective tag
W38	Pre C. Conjunction word
W39	Pre C. Conjunction tag
W64	Post pronoun word
W68	Post Wh word
W69	Post Wh tag
Immediate Pre and Post ... Distance from Cw	
W70	Pre verb distance
W71	Post verb distance
W73	Post adverb distance
W74	Pre preposition distance
W75	Post preposition distance
W76	Pre noun distance
W77	Post noun distance
W78	Pre NNP distance
W80	Pre adjective distance
W82	Pre C Conjunction distance
W83	Post C Conjunction distance
W84	Pre determiner distance
W85	Post determiner distance
W95	Post pronoun distance
W96	Pre Wh distance
W97	Post Wh distance

Table 5: List of Relevant Features at Word Level (D_w)

Next, the list of relevant features at phrase level extracted by the above method is described in Table 6.

Phrase Level Features(D_p)	
Sl	**Name**
P1	Current head Node of the phrase(Ch)
P3	Leaves of the Ch(Cw)
P4	Path from Ch to ancestor Node
P12	has NP as siblings of Ch?
P20	has VP as siblings of Ch?
P26	has COMMA as siblings of Ch?
P27	has STOP as siblings of Ch?
P28	Ancestor Node of Ch(AnCh)
P36	has NP as siblings of AnCh?

Table 6: List of Relevant Features at Phrase Level (D_p)

Now with the help of D_w and D_p we prepared our new training sets as D_{wt} and D_{pt} .Similarly we have prepared our new test sets with these important features as D_{wtest} and D_{ptest} from the text virata parva .

4.1 Features Associativity Analysis

It is observed from the training data sets, D_{wt} and D_{pt}, that some feature or set of features coexists

[1]https://rapidminer.com

with other feature or set of features. This type of relations can be found from the texts very frequently in our sample space. To address this issue we have applied FP-Growth algorithm in word and phrase level. This algorithm calculates all frequent feature/feature set from the data set by building a FP-Tree data structure on the data sets D_{wt} and D_{pt} . Some frequent relations of word and phrase level are given below.

Word Level relations:

Word Level		
Antecedent	**Consequent**	**Confidence**
W20	W14	0.258
W20	W83,W35	0.408
W23	W83,W31,W27	0.352
W20,W35	W14	0.381
W20,W83	W35,W31	0.381
Antecedent -> Consequent		
W14=Immediate pre verb word of Cw W20=Immediate post adverb word of Cw W23=Immediate pre position tag of Cw W27= Immediate pre noun tag of Cw W31=Immediate pre NNP tag of Cw W35=Immediate pre adjective tag of Cw W83=Post C. Conjunction distance from Cw		

Table 7: Features Associativity at Word Level

From Table 7 we can understand that for a distinct context word, Cw,when we identify a value for the feature,W20 in a sentence in the sample space, simultaneously we can find a value for the feature W14 also with a confidence value 0.258.

Phrase Level relations:

Phrase Level		
Antecedent	**Consequent**	**Confidence**
P3	P26	0.261
P26	P3	0.412
P3	P12	0.418
P12	P3	0.743
P26	P12	0.659
P12	P26	0.795
Antecedent -> Consequent		
P3 = Leaves of the Ch(Cw) P12= hasNP as siblings of Ch? P26= hasCOMMA as siblings of Ch?		

Table 8: Features Associativity at Phrase Level

Similarly in the Table 8, when we can observe a value for the feature P3 then P26 is also observed for context word Cw in a sentence of our sample space with confidence value 0.261.

Where X–> Y implies that if X occurred then Y also occurred; X means antecedent and Y means consequent.

4.2 Classification Task

Here we have developed a training model using NeuralNet and KNN classifiers with the help of newly prepared datasets D_{wt} and D_{pt} .Later on we have tested these models using our newly created test sets D_{wtest} and D_{ptest} .At word level NeuralNet has better precision, recall and f-measure than KNN classifier. At phrase level NeuralNet classifier has better precision and f-measure than KNN classifier. On the other hand KNN has better recall value than NeuralNet classifier at phrase level. The precision, recall and f-measure of the two classifiers are explained in Table 9 and Table 10.

Word Level			
Classifiers	**P**	**R**	**F**
NeuralNet	91.84	84.91	88.24
KNN	90.70	73.58	81.25
P=Precision; R=Recall; F=F-measure			

Table 9: Precision, Recall, F-measure on D_{wtest}

Phrase Level			
Classifiers	**P**	**R**	**F**
NeuralNet	79.07	69.39	73.91
KNN	61.67	75.51	67.89
P=Precision; R=Recall; F=F-measure			

Table 10: Precision, Recall, F-measure on D_{ptest}

5 Result Analysis

Both the classifiers performed well in case of classifying the test data sets D_{wtest} and D_{ptest}. NeuralNet classifier has better accuracy than KNN classifier in case of word level and phrase level. The accuracies of both the classifiers for word level and phrase level are discussed in Table 11.

Classifiers	**WAccuracy**	**PAccuracy**
NeuralNet	88.00%	76.00%
KNN	82.00%	65.00%
WAccuracy=Word Level Accuracy		
PAccuracy=Phrase Level Accuracy		

Table 11: Classification Accuracies

The confusion table on accuracies of D_{wtest} and D_{ptest} are given in Table 12 and Table 13 below.

Word Level				
Classifiers	**NN**	**NC**	**CN**	**CC**
NeuralNet	45	4	8	43
KNN	39	4	14	43
C=Character;N=Not_Character				

Table 12: Confusion table for dataset D_{wtest}

From Table 12 we can observe that NeuralNet classifier has correctly classified 88 instances and incorrectly classified 12 instances.Whereas KNN classifier has classified 82 instances correctly and 18 instances incorrectly.

Phrase Level				
Classifiers	**NN**	**NC**	**CN**	**CC**
NeuralNet	42	15	9	34
KNN	28	12	23	37
C=Character;N=Not_Character				

Table 13: Confusion table for dataset D_{ptest}

On the other hand in Table 13 at Phrase level, NeuralNet classifier has classified 24 instances incorrectly and 76 instances correctly.Similarly KNN classifier has classified 35 instances incorrectly and 65 instances correctly.

6 Error Analysis & Observations

It can be observed that NeuralNet classifier has lowest classification error and highest kappa value at word level and phrase level as well. The classification error rate and kappa measure are observed in Table 14.

Classifiers	**Word Level**		**Phrase Level**	
	CE	**K**	**CE**	**K**
NeuralNet	12.00%	0.760	24.00%	0.519
KNN	18.00%	0.643	35.00%	0.303
CE=Classification Error rate; K=Kappa measure				

Table 14: Error and Kappa of D_{wtest} and D_{ptest}

The average absolute deviation of the prediction from the actual value, i.e., Absolute Error of NeuralNet classifier at word level is lower than KNN classifier. Similarly the average of the absolute deviation of the prediction from the actual value divided by the actual value, i.e., Relative Error, and Root Mean Squared Error of NeuralNet classifier at word level significantly lower than KNN classifier. The details are explained in Table 15.

Word Level		
Measures	**NeuralNet**	**KNN**
AE	0.16+/-0.28	0.31+/-0.19
RE	16.83+/-28.09	31.83+/-18.97
RMSE	0.32+/-0.00	0.37+/-0.00
AE=Absolute Error; RE=Relative Error(%)		
RMSE= Root mean Squared Error		

Table 15: Error Analysis of D_{wtest}

At phrase level also NeuralNet classifier has lower Absolute Error, Relative Error and Root Mean Squared Error than KNN classifier. The results are given in Table 16.

Phrase Level		
Measures	**NeuralNet**	**KNN**
AE	0.36+/-0.24	0.41+/-0.25
RE	36.93+/-23.97	41.89+/-25.84
RMSE	0.44+/-0.00	0.492+/-0.00
AE=Absolute Error; RE=Relative Error(%)		
RMSE= Root mean Squared Error		

Table 16: Error Analysis of D_{ptest}

7 Visualization of Co-Occurred Characters

Now, we have measured the co occurrence of all the Characters extracted at word level and phrase level. We analyzed each sentence in our sample space and calculated the co occurrence of each Characters with others. As an example we considered a Character found at word level C="Abhimanyu". In the Figure 3 we have displayed the list of co occurred Characters related to Character C.

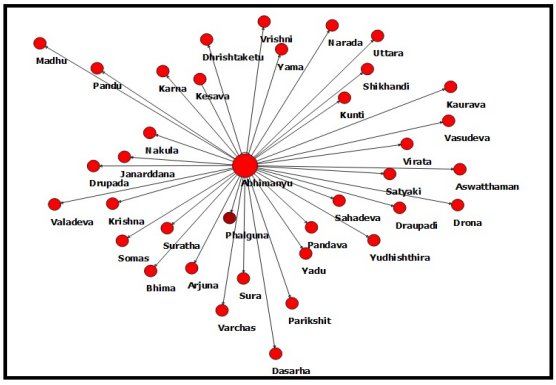

Figure 3: Co occurred Characters w.r.t Character Abhimanyu

8 Conclusion

In this paper our target is to identify the Characters from the Mahabharata. Keeping this in mind at first we have annotated all the Characters present in the sample space. Then we have applied a semi supervised approach with distinct features at word level and phrase level to collect the Characters from the texts and prepared the initial data sets. Then we applied Chi Squares Statistic to find the relevant features from the data sets. According to the relevant features at word and Phrase level we have reshaped our training and testing data sets. Next, we have analyzed the associativity of features using FP-Growth algorithm. Here we found that some features has coexistence with other feature or set of features. Then we developed training models at word level and phrase level with NeuralNet and KNN classifier. Later we have tested our model with our test data and accuracies, precision, recall, f-measure, kappa and different error statistics are observed. As a part of the future work we have planned to increase our sample space with different varieties.

References

Ricardo A Calix, Leili Javadpour, Mehdi Khazaeli, and Gerald M Knapp. 2013. Automatic detection of nominal entities in speech for enriched content search. In *FLAIRS Conference*.

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

Nuno Mamede and Pedro Chaleira. 2004. Character identification in children stories. In *Advances in natural language processing*, pages 82–90. Springer.

Apurba Paul and Dipankar Das. 2017. Identification of character adjectives from mahabharata. In *RANLP*.

Josep Valls-Vargas, Santiago Ontanón, and Jichen Zhu. 2013. Toward character role assignment for natural language stories. In *Proceedings of the Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, pages 101–104.

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

Josep Valls-Vargas, Jichen Zhu, and Santiago Ontañón. 2015. Narrative hermeneutic circle: Improving character role identification from natural language text via feedback loops. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Nees Jan Van Eck, Ludo Waltman, Ed CM Noyons, and Reindert K Buter. 2010. Automatic term identification for bibliometric mapping. *Scientometrics*, 82(3):581–596.