Construction of a Recurrent Neural Network from the Amar Kosha

Dr. Partha Paul
Associate Professor, Department of Computer Science & Engineering,
Sarala Birla University, Ranchi

Kaustav Sanyal
PhD Scholar, Department of Computer Science & Engineering,
Sarala Birla University, Ranchi

Article Info
Page Number: 8958 - 8963
Publication Issue:
Vol 71 No. 4 (2022)

Abstract

In the traditional Sanskrit education system it is considered to be an essential and integral part of learning the grammar is to understand and memorise the complete Amar Kosha. In Sanskrit literary corpus the Kosha Granthas (the written record of the Sanskrit vocabulary) there are many available scripts. Among them Amar Kosha and Halayudha Kosha hold eminent positions. This paper focuses on the construction of a recurrent neural network(RNN) taking the Amar Kosha into consideration to classify the words recorded in it with respect to four prime attributes of the words, viz. the Varga(type), Linga(gender), Antya varna(the ending phoneme of the word) and the paryayavachi (synonyms). Among various schools of grammar of Sanskrit language, the Aindra tradition is the one which dictates a language model on the basis of the morphological structure of the words. It has been observed that Panini's artificial language machine (a model of the Shaiva school of Sanskrit grammar) includes a set of metarule that can reconstruct the lost tradition of Aindra grammar. But in order to implement the set, the words should be modelled in such a way that it can provide a well-defined mathematical model. This work provides the platform with the implementation of RNN on Amar Kosha on which Panini's word making rules can be easily implemented in order to reconstruct the Aindra School of Sanskrit Grammar.

Article History
Article Received*: 25 March 2022*
Revised*: 30 April 2022*
Accepted*: 15 June 2022*
Publication*: 19 August 2022*

Keywords: Sanskrit, Natural Language Processing, Language Model, Recurrent Neural Network, Panini.

Introduction:

The schools of Sanskrit grammar have focused on different aspects of grammar for their language construct. For example, the Barhaspatya school of grammar (the grammar designed by Lord Brihaspati) primarily has taken the roots (dhatu) for the base of the language model [1]. The Aindra school of grammar (the grammar designed by Lord Indra) primarily has taken the morphological structure of the language for its construct [2]. Traditionally before Panini's work, the Aindra school of grammar was widely followed by the linguists and the interpretations of the Vedas and the Vedic texts were completely according to it only. After Panini, all the Vedic and non-Vedic vocabulary have been interpreted through the Ashtadhyayionly [2]. The Aindra school of grammar is considered to be one of the most perfect ones to interpret the Vedic texts. Because of its structural uniqueness the Vedic texts cannot be easily understood by the knowledge of standard construct of the Sanskrit language. In places in Ashtadhyayi, Panini has mentioned the applied exceptions of the Vedic texts for specific vocabulary, like ratri or madhuetc [3]. As the Aindra school of grammar focuses on the understanding of the language only based on its morphological aspects, the interpretations of the Vedic texts for an Aindra grammarian would be far more natural than exceptional. Unfortunately, the root texts and practices of the Aindra school of grammar today is lost and mostly unexplored to the linguists and grammarians.

The Kosha scriptures of Sanskrit literature includes all the possible vocabulary of the language [1]. In that scenario the Amar Kosha and the Halayudha Kosha hold very eminent positions [4]. This paper prepares an RNN out of the vocabulary recorded in the Amar Kosha. The structure of Amar Kosha includes various Vargas that define the type of the word. Furthermore, every word recorded comes with its pre-defined gender (male, female and neuter) and a set of paryayavachi words or the synonyms. The idea behind the work is to prepare a complete word-based network out of all the 11,745 words recorded in the script. The word-based network can be used as the basis of reconstruction of the Aindra school of Sanskrit grammar as mentioned earlier the Aindra school focuses on the morphological construct of the language only. On the other hand, in the domain of neural networks the RNNs have the capability to develop a local memory of itself [5]. Using that idea, the network is constructed in this work.

Methodology:

The words recorded in the script individually are primarily considered as independent nodes without any interconnections. Providing inputs out of those nodes will generate a set of paryayavachi words which are already in the network independently [6]. The interconnection among the nodes will eventually create the dependency with the same weight and biases as defined earlier and will give rise to such a network which has the memory of the previous inputs and thus every word individually will have a set of pre-defined paryayavachi words from the network itself. In the network any word under observation (a current node) will be a function of not only the input and the previous state, but a function of the input and a set of previous states, i.e. ht= f({ht-1, ht-2,….h1}, xt), where ht is the current word and xt is the input. The desired output cluster will be calculated as,

yt= wo+ht

Proposed Algorithm:

Let us define the attributes primarily that will create the basic structure of the morphological network from Amar Kosha. So, for every input X, the classification of the word will be a function of the Varga, Linga, Antya Varna and number of ParyayaVachi or synonyms associated to the word. Mathematically, X= f (A1, A2, A3, A4{n, where n>=0}), where Ai denotes the above-mentioned factors respectively. For A4 we can say

length. A4 = 0 Null Node

>0 Active Node.

The nullification of the nodes without any synonyms will decrease the ranking of the node in the network and will be beneficial for lowering the overall complexity of the network.

The algorithm for training the network will be:

Algorithm 1:

1: **procedure** X= input word

2: **do**

3: move to Hi = ith Hidden Layer

4: feed {AX.1-AX.4} to Hi

5: **if (AX.4>0)**

6: **do**

7: move to Hi+1

8: feed {AX.4.1-AX.4.4} to Hi+1

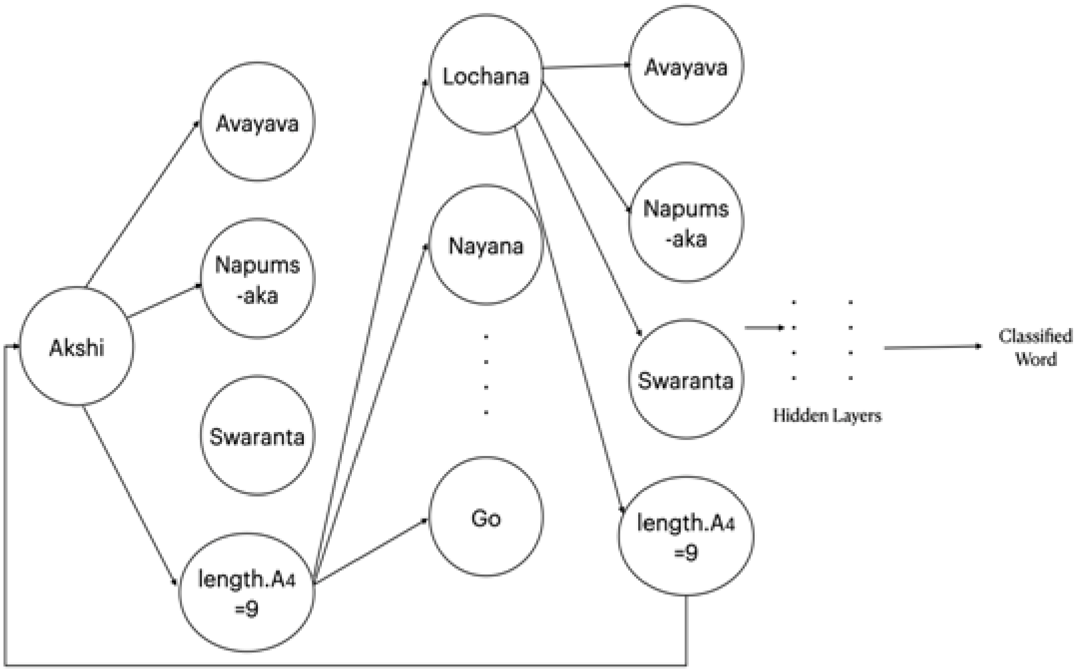
9: **while (AX.4)**

10:**else**

11: X= Null Node

12: **end procedure:**

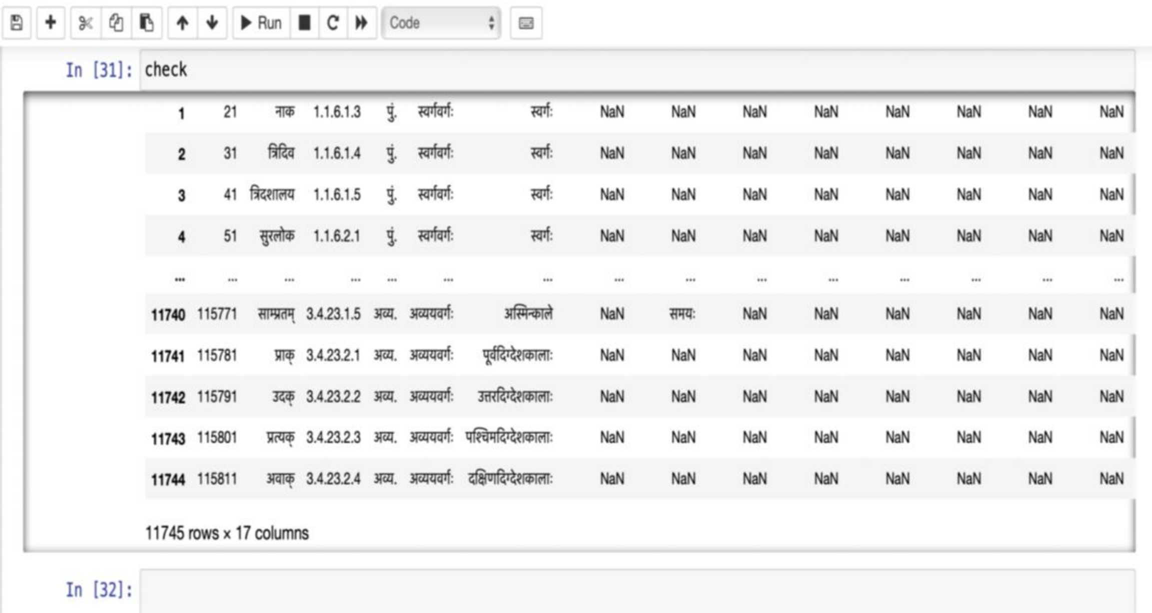
Implementing the algorithm, suppose we have provided the input string as "akshi" (Eye). For the word A1= avayava, A2= napumsaka, A3= akara (swaranta) and A4= {lochana, nayana, netra, ikshana, chakshus, akshi, drs, drsti, go}. So, for length. A4=9 will lead to 9 internal layers from the string "akshi" which themselves will be training the network further with their own A1-A4 attributes. This will give rise to such a recurrent neural network that will contain all the words of Amar Kosha with their own clusters for every attribute.



Example RNN for word "Akshi"

As the network is recurrent in nature all of the nodes associated to the input and attribute A4 will be memorised to provide a strong network for the dictionary. **Result and Discussion:**

The implementation of the proposed algorithm is done with Python programming. From the .csv file of the Amar Kosha, the words have taken as input individually and run through the algorithm. The time complexity for the active network is calculated to be O(n) for 11,745 inputs.



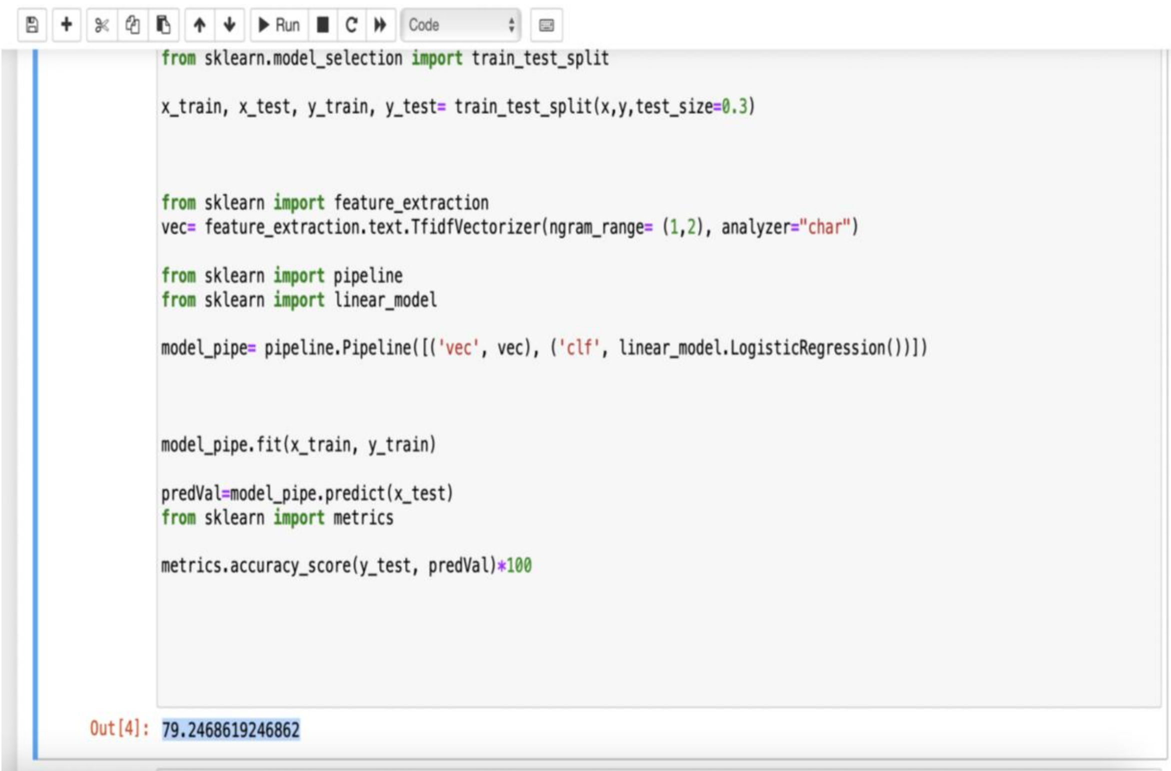
In [31]: check

1	21	नाक	1.1.6.1.3	पुं.	स्वर्गवर्गः	स्वर्गः	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	31	त्रिदिव	1.1.6.1.4	पुं.	स्वर्गवर्गः	स्वर्गः	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	41	त्रिदशालय	1.1.6.1.5	पुं.	स्वर्गवर्गः	स्वर्गः	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	51	सुरलोक	1.1.6.2.1	पुं.	स्वर्गवर्गः	स्वर्गः	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11740	115771	साम्प्रतम्	3.4.23.1.5	अव्य.	अव्ययवर्गः	अस्मिन्काले	NaN	समयः	NaN	NaN	NaN	NaN	NaN	NaN
11741	115781	प्राक्	3.4.23.2.1	अव्य.	अव्ययवर्गः	पूर्वदिग्देशकालाः	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11742	115791	उदक्	3.4.23.2.2	अव्य.	अव्ययवर्गः	उत्तरदिग्देशकालाः	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11743	115801	प्रत्यक्	3.4.23.2.3	अव्य.	अव्ययवर्गः	पश्चिमदिग्देशकालाः	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11744	115811	अवाक्	3.4.23.2.4	अव्य.	अव्ययवर्गः	दक्षिणदिग्देशकालाः	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

11745 rows × 17 columns

In [32]:

Screenshot 1



```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test= train_test_split(x,y,test_size=0.3)


from sklearn import feature_extraction
vec= feature_extraction.text.TfidfVectorizer(ngram_range= (1,2), analyzer="char")

from sklearn import pipeline
from sklearn import linear_model

model_pipe= pipeline.Pipeline([('vec', vec), ('clf', linear_model.LogisticRegression())])


model_pipe.fit(x_train, y_train)

predVal=model_pipe.predict(x_test)
from sklearn import metrics

metrics.accuracy_score(y_test, predVal)*100
```

Out[4]: 79.2468619246862

Screenshot 2

The accuracy of the output desired from the network is calculated to be 79.2468619246862%.

Conclusion and Future Work

The network designed in this work can be treated as a useful tool for the designing of the morphological model of Sanskrit language. The clustering of the words can be of great help to implement the subantaprakarana of Panini's language machine. The implementation of the metarule-set designed to get the output of 21 variations from subantaprakarana, demands the classified input of the data; the Linga and the Varga to be precise. In that scenario this work will be of great help for its self-learning network construct.

Reference

[1] YudhisthirMimamsaka, *Sanskrit Vyakaran Shastra ka Itihas*, BharatiyaPrachya Vidya Samsthan, Ajmer, India

[2] A.C. Burnell, *On the Aindra School of Sanskrit Grammarians*, Basel Mission Book & Tract Depository, Mangalore, India, 1875.

[3] Editor: Dr. Umashankar Sharma Rishi, *VaiyakaranaSiddhantaKaumudi (VaidikiPrakriya)*, ChowkhambaVidyabhavan, Varanasi, 1962.

[4] Editor: Mahamahopadhyay Pandit Shivadattadabhimatha, *Amarkosha*, Chowkhamba Sanskrit Pratisthan, 1984.

[5] Kanchan M. Tarwani, Swathi Edem, *Survey on Recurrent Neural Network in Natural Language Processing*, International Journal of Engineering Trends and Technology (IJETT) – Volume 48 Number 6 June 2017.

[6] Abhyuday N Jagannatha, Hong Yu, *Structured prediction models for RNN based sequence labeling in clinical text*, Proc Conf Empir Methods Nat Lang Process. 2016 Nov; 2016: 856–865, doi: 10.18653/v1/d16-1082.