Semiconductor Optoelectronics
ISSN: 1001-5868

SKIP-GRAM VECTORISATION OF THE SEMANTIC RULES OF PANINI FOR ARTIFICIAL NEURAL NETWORK FOR SANSKRIT GRAMMAR

Kaustav Sanyal
PhD. Research Scholar, Department of Computer Science and Engineering, Sarala Birla University, Ranchi, Jharkhand

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

Abstract: The Ashtadhyayi composed by Maharshi Panini is composed of almost 4000 semantic rules, each corresponding to a definite task and in some cases other uses as well. This idea holds resemblance to the idea of word embeddings in standard Natural Language Processing methods. With the implementation of the word2vec approach, this paper focuses on the vectorisation of the meta rules of the Paninian framework through an artificial neural network (ANN) to find out the accuracy of the proposed associated rules by Panini from the machin learning perspective in computational sanskrit. In this work with the implementation of the skip-gram method of word2vec vectorisation technique the Paninian rules are used to build an ANN to study the behaviour of the rules and their association in an artificially generated language model in machine learning.

Keywords: Panini, computational sanskrit, Ashtadhyayi, word2vec, natural language processing, skip-gram, artificial neural network, machine learning.

Introduction:

The meta rules proposed by Maharshi Panini in his groundbreaking grammatical masterpiece Ashtadhyayi has given rise to not only a complete structural basis of Sanskrit language but also to an artificial language model, that can be analysed and structured in a mathematical manner. Thousands of manuscripts that have been composed in Sanskrit language have not been properly decoded yet because of the lack of understanding of the language itself, though the computation of Sanskrit language is comparatively simpler than the other languages of the world which have been successfully computed in standard Natural Language Processing (NLP) approach [1]. But in order to implement the artificial language model of Panini computationally there are several challenges that are needed to be overcome, such as,

1. Computational complexity of the meta rules
2. Lack of understanding of the structural basis of Panini’s framework
3. Lack of understanding of the models proposed in the work itself
4. Failure to compute the mathematical interpretation of the rules by Maharshi Patanjali and Maharshi Katyayana [2]

5. Failure to mathematically interrelate the rules and so on.

This work has focused on overcoming one of the several challenges of Paninian framework and that is construction of an interrelation network or model out of the sey of rules proposed. For the work we have considered the standard approach of word embeddings in the domain of NLP, implementing the skip-gram approach. As the word embedding approach is the analysis of the text and representation of the words used in it from a text corpus, we have taken the liberty to do the same for the meta rules themselves. Today when we are living in such an age when we can design and create a machine whose outputs with errors can be fed back to the same network to train the system in a better manner, we have implemented the idea of artificial neural network and implemented on the Paninian framework to study the output and the mathematical accuracy of the work.

Background Study:

Panini's language model can be considered to be the most modern rule book for Sanskrit grammar. Apart from Panini there are several schools of grammar in the Sanskrit language who explain the language construct in different manners. In Panini's work we can see the compilation of almost all the other traditions, as he acknowledged himself the rules he took from Brihaspati or Katyayana and others. But the marvel of Ashtadhyayi is the simplicity and the simpler understanding of the language providing a very strong artificial language model of its own. We are referring to the Paninian framework as an artificial language model for the following reasons:

1. The meta rules proposed by Panini are very similar to the semantic rules that are used to form an artificial neural network (ANN).
2. Traditional teaching-learning approach of Ashtadhyayi includes memorisation of all the rules with their index number intact, which is very similar to assigning weights to the input nodes.
3. Several rules refer to other rules in action and implementation to provide a specific output, where the rules outbound from the root rule act like the hidden layers of an ANN.
4. The final output (may that be a word or a sentence) are rechecked through the complete system again which will not only include the meta rules of Ashtadhyayi but Panini's other related works as well, like the Dhatupatha, Phit Sutras, Paniniya Shiksha etc. just like the post optimal analysis for an output of an ANN.

For more similar reasons we have considered it to be absolutely credible to refer to Panini's work and the language generated from it as an Artificial Language Model.

On the other hand, in NLP, the word embeddings play a very crucial role to construct a predictive language model. The embeddings of the associated words with the implementation of a series of word vectors, run through an Activation function allows us to predict the association of other words to the input vector [3]. Between continuous bag of words (CBOW) and skip-gram approach of word embedding techniques we have chosen skip-gram approach. To make an ANN out of the Paninian framework primarily we need to vectorise the rules themselves. For the rules are distinct and suggest very specific action in the language model they cannot be treated with the CBOW approach [4]. The skip-gram approach for its

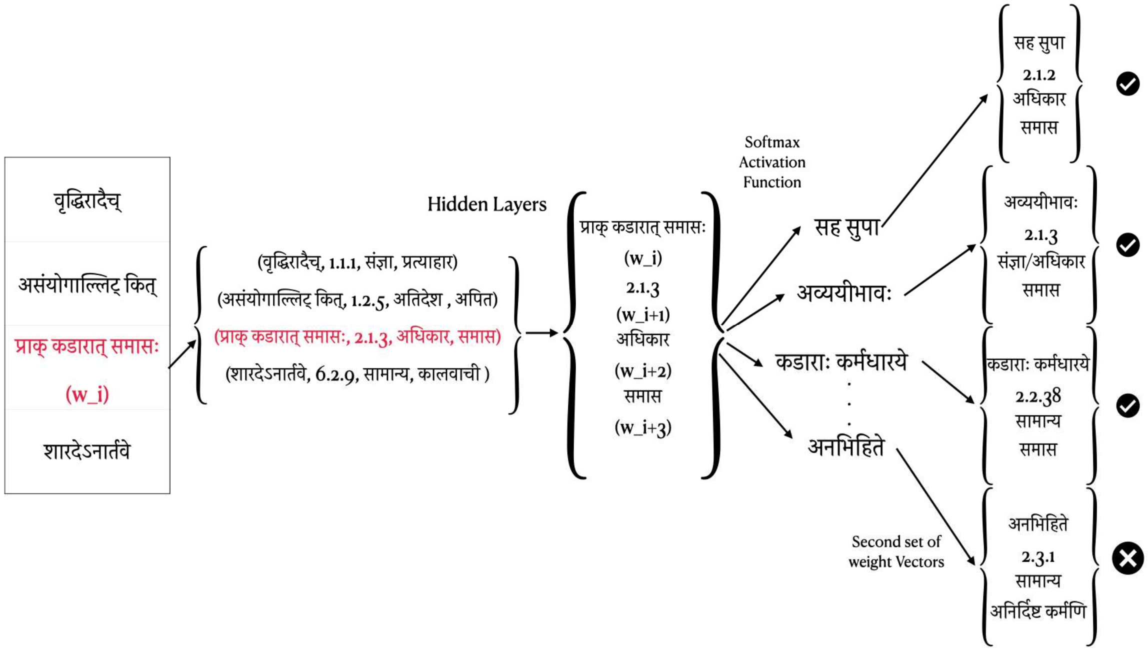

Fig: Skip-gram flow chart of sutra prediction

consideration of the individual words to predict the text through the ANN serves the purpose of computing the distinct rules through the ANN to find out the associated rules in the Paninian framework. The loss generated from the computation will be back propagated to the primary hidden layer for further training and deep neural network computation.

Proposed Algorithm:

Unlike the standard approach of the skip-gram approach of word embedding in NLP, we have considered the aspects associated with the meta rules of Panini [5]. The difference in the applied area of this work and the standard skip-gram approach is the selection of training corpus. Where the standard skip-gram algorithm considers a definite word in a sentence to vectorise the other words in association of its context from a large number of corpora, we have considered the meta rules themselves of Panini [6]. The rules proposed in the Ashtadhyayi come with a series off pre-defined attributes like, indexing, categorisation, application domains etc[7]. For example, if we take the rule प्राक् कडारात् समासः, it comes with an indexing i.e. 2.1.3 (where 2 is the chapter number, 1 is the module number and 3 is the number of the rule), category is अधिकारः and domain is up to the end of the second module[8]. If the rule is w_i, the index, category and domain will be considered as w_i+1, w_i+2 and w_i+3. The flow chart is as follows:

Now the input rule after going through the softmax activation function is expected to give an output (with maximum vector value of 1) of the predicted other rules associated to the input rule[9]. The activation function can be defined as,

$$softmax(predictedRule_n) = \frac{e^{predictedRul\quad n}}{\sum_{i=1} e^{predictedR\quad i}} \dots \text{(Equation 1)}$$

Result and discussion:

The system has been trained with 3985 rules of Panini with their associated attributes. The initial input layer consisted the rules moved through the look up table of the same with their attributes as a collection of 3985 number of distinct sets. This moves forward to produce a set of a definite set of word embeddings for a specific rule which further moves through the softmax activation function. This step produces a number of predictive output rules with distinct word embedding sets.

The outputs are the vectors supposed to be on a 2D plane. Their distance from the input rule is now expected to be calculated to be identified as the final output. For this purpose, between cosine and euclidean distance we have chosen euclidean distance [10]. The reason behind the choice is to keep the output vectors as distantly distinct and differentiable as possible. Euclidean distance serves the purpose. The reason to keep the vector points in such a manner is based on the fact of the degree of word loss in each iteration. The degree of word loss in every iteration is back propagated to the look up table embedding for better learning for the rest of the rules.

$$d(firstVec, secVec) = \sqrt{\sum_{i=1}^{numberofdimensions} (firstvec_i - secVec_i)^2} \; \ldots \text{(Equation 2)}$$

$$Loss = -\sum_{i=1} rule_i * ln(targetRule_i) \; \ldots \text{(Equation 3)}$$

Conclusion:

The algorithm works with its own limitations, such as

1. It does not respond to the phit sutras.
2. The rules from dhatu patha are not included in the training model [11].
3. Incapable of identifying application of the rules in a word or sentence produced following the rules themselves.

As future work, these areas can be covered to make a more efficient ANN for the language machine of Panini [12]. Moreover, the skip-gram vectorisation of the Paninian language model can be used as a useful tool for the domain of computational sanskrit. For machine translation and deep learning in sanskrit grammar this work can be a useful tool.

Reference:

[1] Subhash C. Kak, The Paninian Approach to Natural Language Processing, International journal of Approximate Reasoning, 1987; 1:117-1130.

[2] J.F. Staal, A Reader on the Sanskrit Grammarians, MIT Press, Cambridge, Mass., 1972.

[3] Manish Kumar, Mohit Dua, Adapting Stanford Parser's Dependencies to Paninian Grammar's Karaka relations using VerbNet, Second International Symposium on Computer Vision and the Internet (VisionNet'15), doi: 10.1016/j.procs.2015.08.033.

[4] Rashi Kumar; Vineet Sahula, Word Translation using Cross-Lingual Word Embedding: Case of Sanskrit to Hindi Translation, 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP), DOI: 10.1109/AISP53593.2022.9760564.

[5] Mohammad Taher Pilehvar , Jose Camacho-Collados, Embeddings in Natural Language Processing:Theory and Advances in Vector Representations of Meaning, Springer Cham, ISSN-1947-4040, DOI: https://doi.org/10.1007/978-3-031-02177-0.

[6] Kumar Saurav, Kumar Saunack, Diptesh Kanojia, Pushpak Bhattacharyya,
"A Passage to India": Pre-trained Word Embeddings for Indian Languages, Proceedings of the 1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020), pages 352–357 Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020.

[7] Editor: Dr. Umashankar Sharma Rishi, *Vaiyakarana Siddhanta Kaumudi (Vaidiki Prakriya)*, Chowkhamba Vidyabhavan, Varanasi, 1962.

[8] Ashtadhyayi.

[9] Wei Di, Anurag Bhardwaj, Jianing Wei, Deep Learning Essentials, Packt Publishing, ISBN: 9781785880360, 2018.

[10] Leo Liberti, Carlile Lavor, Euclidean Distance Geometry: An Introduction, Springer Cham, Softcover ISBN: 978-3-319-86934-6, 17 May 2018.

[11] Translation by Srisa Chandra Vasu, The Ashtadhyayi of Panini, Indian Press, Allahabad, 1891.

[12] Hugh Cartwright, Artificial Neural Networks, DOI: https: doi.org10.1007/978-1-0716-0826-5, Publisher: Humana New York, Softcover ISBN
978-1-0716-0828-9, Edition 3, Published: 18 August 2021.