Computational Sanskrit

# Word embedding for order-agnostic languages (word2vec for Sanskrit)

## Order-agnostic languages

In languages like English, meaning of a sentence is dependent on the order of the words in it. E.g. “John loves Mary” has different meaning than “Mary loves John”. But in more evolved/formal languages, like Sanskrit, the meaning is order-agnostic. Because each word is suffixed (declension? Vibhakti) so that its purpose is clear. John is changed to its nominative case form (prathamā) and Mary is changed to her accusative case form (dvitīyā). So, wherever they appear in the sentence, meaning remains unchanged.

## Working of wored2vec

It’s a single layer neural network with random weights to start with. A sentence is examined by a running window of n words (n-gram) with focus word in the middle. Value of the focus word is determined based on the values of the neighbouring word values. This results in measuring likelihood of words being associated with the focus words. Thus, similarity or associations can be computed easily.

But that’s not the case with Sanskrit. As the words in a sentence are order independent, word associations computed in traditional manner are not correct.

## Proposed S-gram approach

Instead of running window on n-grams, for Sanskrit, we need to look at one full sentence at one time. Give equal values to all the words in it. Move to the next sentence. So, it’s a S-gram (Sentence at a time) approach.

## Next steps

* Find Sanskrit data
* Modify word2vec implementation
* Test the proposed S-gram hypothesis

# References

1. DCS - Digital Corpus of Sanskrit <http://kjc-sv013.kjc.uni-heidelberg.de/dcs/index.php>