A mini-review on assessing the relationship between Sanskrit and Computer programming languages for recounting its significance in Artificial intelligence

Abstract:

Sanskrit is a valuable, unambiguous ancient Indian (Hindu) language using which an abundance of Hindu Scriptures and knowledge has been produced. Having its grammatical structure enriched/revised by a sage named Maharishi Panini, Sanskrit's full potential in the world of logic and computing still hasn't been discovered. Hence, this review can be considered an Ode on the usefulness of Sanskrit in Natural Language Processing (NLP), its grammar's syntactic similarities with computer programming languages and the hope it provides for space communication and making graphics processing units (GPUs) faster.
The reviewed data can be insightful by providing the contextual and historical background of Sanskrit and certain Natural Language Processing and computer programming concepts.

Introduction:

Sanskrit - referred to as "Dev Vani" i.e the language of gods, also considered as the mother of all languages by many, is said to be created by Lord Brahma who provided the ancient Hindu Sages with it. Later, a renowned grammatician - Maharishi Panini (520 BC - 460 BC) born in Shalatula (a town near the Indus river), had worked on several other grammatical works before deciding to "confine" Sanskrit by giving it a more comprehensible structure and form by changing its morphology, phonetics, etc[1]. Panini's grammar is comprised of Sutras (now commonly known as "Strings" in modern-day programming languages) that have special characteristics to give depth and complexity to the words that are formed using them. Moreover, there are 14 main Sutras that Panini described sounding exactly like the 14 rhythms he heard from Lord Shiva's drum. Paninian grammar's detailed description lies in his book Astadhyayi which will be also be explored in the following paper.
It is important to emphasize the potential of Paninian grammar to help better comprehend its value and scalability. "Where Sanskrit meets Computer Science"[2] - written to prove the compatibility of Science and Shastras, reviews the work of Dr P. Ramanujan on "Computational rendering of Paninian

grammar" - includes a beautiful statement made by Dr P . Ramanujan: "if a language has many meanings for a word, it is ambiguous, but when Sanskrit has many meanings for a word, it is rich!". Moreover, it is argued that there is some "unconscious knowledge " that makes the speaker fluent in a particular language, this knowledge is an intuition about a language's functionality and flexibility that is acquired after years of experience using the language.
Dr Ramanujan portrays the "predetermined" and "derivable" nature of Sanskrit's grammar and elaborates that a word always has "a base meaning, a suffix meaning, and a combination meaning" wherein the base will always be constant and remain the same and the suffixes will be "variables" i.e suffixes provide more functionality and complexity to the language, making Paninian/Sanskrit grammar very similar to present-day programming language syntaxes.
Additionally, Walter Euegene Clark, a Professor of Sanskrit at Harvard University, after translating Aryabhatta's Aryabhatiya to English, praised Paninian grammar by calling it "the earliest scientific grammar in the world, and also one of the greatest." He mentioned that the "Indian study of language was as objective as the dissection of the body by an anatomist" portraying the usefulness and precision of Paninian grammar making it all the more favourable to be used as a programming/logic-based language.

Another intriguing detail regarding Paninian grammar was revealed in "Sanskrit: the first programming language?"[3] by Tom Goldenberg to prove that Sanskrit grammar has a fresh and unique way of viewing a language that is worthy of appreciation. It is said that Sanskrit grammar has the same characteristics as a programming algorithm, as "Algorithms are the poetry of computation. Just like the verse, they can be terse, allusive, dense, and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing ". The potency of Paninian grammar can be realized by the fact that the Comprehensive English grammar takes up 2000 pages to be fully written down whereas Paninian grammar only takes 1% i.e 20 pages!; the reason for Paninian grammar to be short and crisp is that Sanskrit is to be learned by verbal recitation and hence it has to be easy to memorize.

Objective:

The main objective of this review is to ignite lost interest and appreciation in the ancient Hindu language Sanskrit by providing Historical, and technical context regarding Sanskrit and the fields of Computer Science that it has proved to be useful in (Natural Language Processing, computer programming languages and graphics processing units).
Moreover, this paper is written with the aim of assisting and encouraging further research about Sanskrit's use in the development of technology and space communication.

The Literature review process:

The review process started with selecting an overall theme for research then a particular topic with a question associated with it was narrowed down and specified. Then, a thesis was developed and sources of information that supported the thesis were searched and selected. Reliable websites like Research Gate, IEEE, etc were used to take sources from and it was made sure that the "AND", "OR", and "NOT" operators were used while searching for articles online so the search engine could understand what information was specifically being looked for.
Lastly, extremely lengthy, vague articles that did not describe the research topic in detail were avoided and it was made sure that all the articles/research papers being referred to were read from the abstract to the conclusion to effectively understand their underlying purpose/overall gist.

Sanskrit and computer programming languages

Prior to understanding the link between Sanskrit and programming languages, one needs to understand the concept that is commonly known as Object-Oriented Programming (OOP). Object-Oriented Programming is a manner of programming which is associated with the use of Objects, Methods/functions, and Classes. To understand this as a non-abstract entity, the Class is the world in which all the programs reside, classes have subclasses just like the Earth has countries that have cities that have districts, etc; Objects are entities like humans that can act as variables or constants that are used for the inputs and outputs of a program; the methods/functions are blocks of code that process the information provided by the Objects and output results in the Class.

Similar to the syntax of modern-day programming languages, Paninian grammar also has had a very strict grammatical rulebase. Moreover, its grammar makes Sanskrit unambiguous(not being open to more than one interpretation) - which makes it useful if used as an artificial language, as certain words will act as command terms that only work when specified. The standard method for analyzing Sanskrit text is described to be in the following order: <word><base><form><relation> moreover each term in the order is similar to that of any present-day language. Moreover the "Rulebase of Sanskrit" includes the Samjna, Adhikara, and Paribhasha sutras. which in simple terms are certain conditions/grammatical requirements that have to exist for the sutras to get interpreted and triggered just like certain operators and predefined statements in an integrated development environment (IDE) of a programming language.

The article "An analysis of current trends for Sanskrit as a computer programming language"[4] provides additional, concrete evidence supporting the claim that Sanskrit is similar to a computer programming language. Using Sanskrit to run computers will highly improve their processing speeds and will also make computer programming languages more "flexible, logical and compact".
The 14 Maheshwar sutras are also touched upon in the abstract which are said to be responsible for the creation, explanation, and exploration of Sanskrit's clear and discrete grammar that has "few exceptions".

The Maheswar sutras:

1. a i u Ṇ	१. अ इ उ ण् ।ൎ
2. ṛ ḷ K	२. ऋ ऌ क् ।ൎ
3. e o Ṅ	३. ए ओ ङ् ।ൎ
4. ai au C	४. ऐ औ च् ।ൎ
5. h y v r Ṭ	५. ह य व र ट् ।ൎ
6. l Ṇ	६. ल ण् ।ൎ
7. ñ m ṅ ṇ n M	७. ञ म ङ ण न म् ।ൎ
8. jh bh Ñ	८. झ भ ञ् ।ൎ
9. gh ḍh dh Ṣ	९. घ ढ ध ष् ।ൎ
10. j b g ḍ d Ś	१०. ज ब ग ड द श् ।ൎ
11. kh ph ch ṭh th ca ṭ t V	११. ख फ छ ठ थ च ट त व् ।ൎ
12. k p Y	१२. क प य् ।ൎ
13. ś ṣ s R	१३. श ष स र् ।ൎ
14. h L	१४. ह ल् ।ൎ

Basic information about the Sutras is contained in Panini's book "Astadhyahi", and it is to be heavily stressed how valuable Sanskrit as a language is due to it being spoken and studied for over 1000 years and the potency of this language hasn't yet been fully discovered.
Furthermore, the article [4] names the two important pieces of research done: First- Rick Briggs' work that focuses on semantic nets, Shastrik Sanskrit, and equivalence that makes Sanskrit fit for being used as an Artificial language. Second - Shashank Mani Tripathi's work explains the significance of Sanskrit when used in space communication as a computer programming language.
Additionally, the review [4] shows the statistics of how often work/research was conducted on Sanskrit - peaking in 2006, 2010, and 2013 - showing that more work is needed/ the lack of research.
Lastly, it concludes by stating: Since 1985 studies have proven that Sanskrit is beneficial for natural language processing (NLP).

Other works similar to the ones above also argue that there are indeed numerous similarities between Sanskrit and Computer Science. For example, "Ashtadhyayi - An experimental approach to Enhance Programming Languages and Compiler Design Using Panini's Grammar"[5] refers to the origin of Sanskrit, calls it a very logical language and refers to the creator of Sanskrit grammar (Maharishi Panini) as the first programmer to ever exist. Moreover, by the use of the 4000 efficient Sutra's - the article claims that graphics processing units (GPUs) can be enhanced and made much more efficient by the use of a Panini-inspired compiler for C (a programming language). This work [5] mentions the significance of Vedic mathematicians and scientists, proving their credibility and prowess, also defines Sanskrit to be "perfect" and calls it the language of gods due to it being spoken and studied for centuries. Panini has compiled all his work in his book "Astadhyayi"' which consists of 8 chapters that are further subdivided into quarters, the book is also said to use and explain grammar in the form of a mathematical function that integrates 1700 core elements like nouns, verbs, etc. Astadhyayi is studied using the Dathypatha and Ganapatha which are classes/sublists of verbal roots and morphs. It is further mentioned that Asthadyayi describes Sanskrit's phonology, morphology, and syntax by the use of its 4000 sutras that define its structure as a rule-based system. Every sutra is said to be a reference to a rule with theorems and meta theorems and each sutra is said to have only three words. Then the article gives an example of the types of suras and explains that they consist of prefixes, suffixes, recursion, and context (all these characteristics are inherent in modern-day coding and programming languages).

The methodology used for the experiment [5] :

1. Understanding the sutra style of grammar and pondering over current programming languages like C and CUDA-C.
2. Design - to make a compiler using YACC for the new Panini-inspired grammar and to develop a matrix multiplication program using it.
3. Validation - to compile and run a program to compare it between the normal and the Panini compiler, and to analyze the results obtained.

Lastly, the conclusion was that Paninian grammar's optimizations caused the compiler to speed up by 40-50%, and Panini was proved to be the first programmer.

Adding to the works above, the article "Sanskrit Computational Linguistics"[6] mentions that Panini has culminated thoughts and grammatical analysis into his ancient book "Ashtadyayi" the studies of which have gained pace due to the modern information theory. It is also to be understood that the importance and relevance of Ashtadyayi have become threefold since it helps understand Sanskirt's grammar which helps understand the basic framework of other commonly spoken languages as well

since most new languages were derived from ancient ones. Sanskrit having less than 4000 Sutras in total allows it to act as programming for humans instead of it being a programming language for computers. Lastly, the work [6] mentions that French computer scientist and mathematician - Gerard Huet called Panini the father of Informatics due to Sanskrit having a rich structure and It is said that Sanskrit was called the "Lingua Franca" of the world of intellectuals.

In the latter half, the research [6] mentions the creation of a Hindi - Sanskrit parser that has been developed to analyze Hindi sentences as they were derived from Sanskrit. A tagged corpus is also being developed for Indian languages that are based on Sanskrit grammar.
Furthermore, the article says Panini mentions that only Karaka relations can be extracted and the themantics cant as one needs to understand knowledge for it. Examples to prove that Karakas are the basic "code" for any sentences formed using Panini's grammar are then given in the work [6].
The only difference between English and Indian languages is that English doesn't have an accusative marker at the phonemic level which causes information of an answer to a "yes-no" question to be lost in the process. If "faithfulness" is to be achieved, the "naturalness" /the natural beauty of the source language is sacrificed.

Then it is explained how the "Anusaraka" i.e the language processor can help a "serious reader" interpret the meaning and essence. Moreover, the difference between Machine translation (MT) and Anusaraka is explained:

1. Anusaraka Modules with high reliability are favoured over ones that are less reliable so the user is most likely to get accurate results first.

2. MT allows the user to hide undesirable information and also gives the user contextual help - providing the user with total control over the process.

The paper [6] later briefs the reader about the working of Sanskrit - Hindu Anusaraka by the use of outputs from it, and lastly, it is said that the act of developing computational tools for Sanskrit has been made easier by the use of Ashtadhyahi, and it is a better alternative to applying the rules of Paninian grammar to other languages. It is also described that finding computational tools for Sanskrit is cumbersome despite the availability of abundant literature because Sanskrit Scholars rarely turn towards computer science, and then the word-formation in Sanskrit that involves the interchanging of constant prefixes and suffixes is explained.

"Panini's Grammar in Computer Science"[7] mentions that Sanskrit grammar can act as a language generator for languages like Chinese or Thai.Secondly, it is explained that Sanskrit grammar is similar to computer programming languages as it is unambiguous i.e there are very specific grammar rules and structures, moreover, Panini's rules are based on the microanalysis of each syntactic structure/format.
Then certain common properties of Paninian grammar and Computer Programming languages are described, namely:

1. Databases: The databases Panini used were:
    1. Aksharasam Amanaya - the 14 Shiva Sutras
    2. Sutrapatha - the 4000 Sutras
    3. Datupata - the 2014 verb roots
    4. Ghanpatha - contains primitives and Avayayas

2. Recursion: to complete some padas, he took them from previous Sutras and “called the sutra in the sutra” which is very similar to modern-day recursion.

3. Arrays: the 14 Shiv/Maheshwar sutras were written and ended with a /0 at and just like ‘/n’ is a comment line ‘/0’ is also doesn’t have an output result, so by visualizing certain facts we can start to see the similarities between them.

4. Inheritance: Panini’s grammar has classes like “Javika” which have two other base classes and they inherit meanings from each from their parent class.

5. Polymorphism: an important feature of object-oriented programming, basically means one name having multiple meanings and Panini took a similar approach when he declared the Bahubriha Samas.

Sanskrit and Natural Language Processing

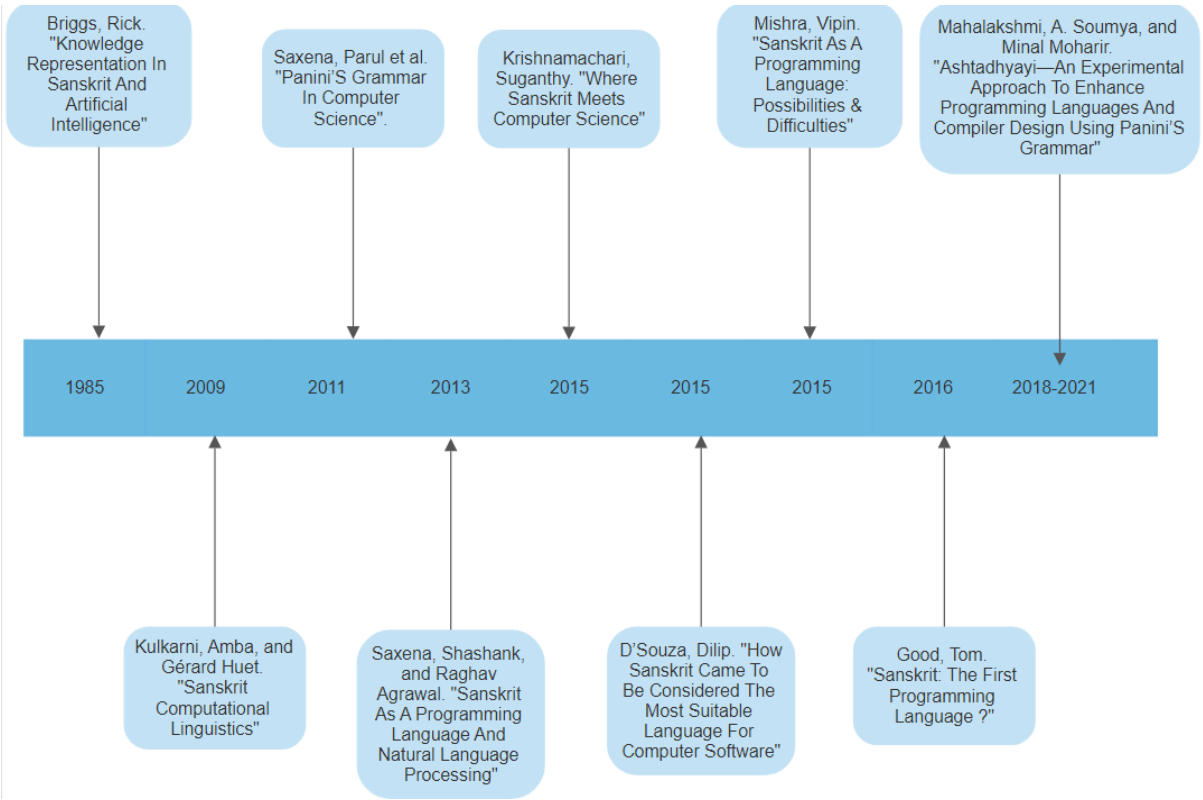
Now that the fundamental relation between Sanskrit and computer programming languages has been discussed, the relation between Sanskrit and Natural Language Processing (NLP) is elaborated .

NLP is a field of Artificial Intelligence(AI) that gives computers the ability to comprehend any commonly known and spoken language like English in the form of text or spoken sentences - just how humans converse with each other. It is a combination of computational linguistics and analysis of the fundamental *Rule Base* of a language. An NLP provides the computer with functionality that can understand/respond to users, summarise large amounts of data and output it to the user’s device. Common examples of Natural Language Processors are Global Positioning Systems (GPS), Google/IOS/Samsung assistants on your mobile devices, and Text-To-Speech software [8].

NASA scientist Rick Briggs’ research published in 1895[9] explores how a natural language can be used as an artificial language in AI and computer programming and argues that Sanskrit is best suited for this. The paper is begun by exploring the semantics of Sanskrit grammar by the use of English words and sentences where he breaks down simple sentences into main events, objects, verbs, identifiers, and specifications - this is done by making a Top-Down design showing the importance of certain words in the sentence in a hierarchical manner. He then describes Shastric Sasnkrit’s sentence formations and how Sanskrit grammarians would paraphrase a simple sentence - making it longer than before but achieving more preciseness and making it more grammatically correct, they do this by re-writing every sentence expressing an action by the means of both verbs and a set of “auxiliaries”. Lastly, Briggs talks about the “Equivalence” of the sentence analysis previously explained and describes the cause - event/result - action - object similarities of Sanskrit grammar to how the ideal AI compatible natural language would be by showing certain sentences, suffixes, prefixes, and primitives. Additionally, “Sanskrit as a programming language and Natural Language Processing”[10] aims to develop a “Dependency parser”(a parser is a code compiler that is used to break and divide data in a way that allows it to be managed and analyzed effectively) by the use of the Sanskrit Rulebase system, concludes by mentioning that they managed to successfully demonstrate the parsing of Sanskrit using their approach of using the full semantics instead of relying on derived stages that indicate what is most appropriate to do so. Moreover, the article ends with the mention of

Vaakkriti which is a method to try and improve the work done in the article and to generate a Natural Language Processor successfully.

Timeline of noteworthy articles written(1985-2021)



Conclusion:

Even though there hasn't been much awareness regarding the potential of Sanskrit at present time, the successful implementations of Paninian grammar in the field of computer science have opened new gateways of technological development, where hardware resources like CPUs and GPUs can possibly be used more efficiently, and where Sanskrit could be used for space exploration even.
These successful implementations of Paninian grammar help reignite interest and appreciation for Sanskrit and its architecture and also support my thesis that Computer Science has/can be benefited from Sanskrit and its grammar.

Author: Adhiraj Singh Saharan
School: Neerja Modi School
Class: International Baccalaureate Program grade 11
Email: ahiraj1180@gmail.com

References

1. "Panini - Biography". Maths History, undated,https://mathshistory.st-andrews.ac.uk/Biographies/Panini/.

2. Krishnamachari, Suganthy. "Where Sanskrit Meets Computer Science". *The Hindu*, 2015, https://www.thehindu.com/features/friday-review/where-sanskrit-meets-computer-science/article7061379.ece.

3. Goldenberg, Tom. "Sanskrit: The First Programming Language ?". *Medium*, 2016, https://medium.com/@tomgoldenberg/sanskrit-the-first-programming-language-d8647753217f.

4. Tiwari, Manish. "An Analysis Of Current Trends For Sanskrit As A Computer Programming Language.". *Researchgate*, 2019, https://www.researchgate.net/publication/339374895_AN_ANALYSIS_OF_CURRENT_TRENDS_FOR_SANSKRIT_AS_A_COMPUTER_PROGRAMMING_LANGUAGE.

5. Mahalakshmi, A. Soumya, and Minal Moharir. "Ashtadhyayi—An Experimental Approach To Enhance Programming Languages And Compiler Design Using Panini'S Grammar". *Springer Link*, 2018, https://link.springer.com/chapter/10.1007/978-981-10-8633-5_1.

6. Kulkarni, Amba, and Gérard Huet. "Sanskrit Computational Linguistics | Springerlink". *Link.Springer.Com*, 2009, https://link.springer.com/book/10.1007/978-3-540-93885-9.
7. Saxena, Parul et al. "Panini'S Grammar In Computer Science". *Https://Www.Researchgate.Net/*, 2011, https://www.researchgate.net/profile/Vinay-Saxena/publication/216868723_Panini%27s_Grammar_in_Computer_Science/links/09e415093c9c26937c000000/Paninis-Grammar-in-Computer-Science.pdf.

8. Education, IBM. "What Is Natural Language Processing?". *Ibm.Com*, 2020, https://www.ibm.com/cloud/learn/natural-language-processing.

9. Briggs, Rick. "Knowledge Representation In Sanskrit And Artificial Intelligence". *AI Magazine*, 1985, https://ojs.aaai.org//index.php/aimagazine/article/view/466.

10. Saxena, Shashank, and Raghav Agrawal. "Sanskrit As A Programming Language And Natural Language Processing". *Ripublication.Com*, 2013, http://www.ripublication.com/gjmbs_spl/gjmbsv3n10_14.pdf.

11. Mishra, Vipin. "Sanskrit As A Programming Language: Possibilities & Difficulties". *Ijiset.Com*, 2015, http://ijiset.com/vol2/v2s4/IJISET_V2_I4_176.pdf.

12. Education, IBM. "What Is Natural Language Processing?". *Ibm.Com*, 2020, https://www.ibm.com/cloud/learn/natural-language-processing.

13. "Parser - Javatpoint". *Www.Javatpoint.Com*, https://www.javatpoint.com/parser.

14. Pati, Satavisa. "Adoption Of Sanskrit By NASA Aims To Change The Language Gap". *Analyticsinsight.Net*, 2021, https://www.analyticsinsight.net/adoption-of-sanskrit-by-nasa-aims-to-change-the-language-gap/.

15. "Sanskrit Alphabet Shiva Sutras (Maheshvara Sutrani) | The Matheson Trust". *Themathesontrust.Org*, https://www.themathesontrust.org/library/shiva-sutras.