EasyChair Preprint

№ 10186

Natural Language Processing for Sanskrit Language Using Logic Programming

Venkata Subba Reddy Poli

May 17, 2023

Natural Language Processing for Sanskrit Language using Logic Programming

P. Venkata Subba Reddy

Abstract—Indian languages have long history and Sankrit is the first language. Panini 's was the first to define Grammar for Sanskrit language with about 4000 rules in sixth century BC . The Natural Languages are only possible to processes with the English character set. It is possible to process other Indian languages by translating in to English character set. The fundamental aspect of Natural language processing is Knowledge representation The Panini's Sanskrit Grammar shall be represented in Predicate Logic. In this paper Sanskrit grammar is represented in First Order Predicate Logic. The Sanskrit Language character set is translated into English character set using Om Transliteration for Sanskrit Language Processing Sanskrit language is processed with the Logic Programming using English character set.

Keywords—Panini's Sanskrit Grammar, Sanskrit Language processing, Logic programming, SWI-Prolog, Om Transliteration.

I. INTRODUCTION

Here is some discussion of the brief history of Sanskrit Language[14, 15]. A particularly important development in the history of Indian science that was to have a profound impact on all mathematical treatises that followed was the pioneering work by Panini (6th C BC) in the field of Sanskrit grammar and linguistics. Besides expounding a comprehensive and scientific

Theory of phonetics, phonology and morphology, Panini provided formal production rules and definitions describing Sanskrit grammar in his treatise called Asthadhyayi. Basic elements such as vowels and consonants, parts of speech such as nouns and verbs were placed in classes. The construction of compound words and sentences was elaborated through ordered rules operating on underlying structures in a manner similar to

Today, Panini's constructions can also be seen as comparable to modern definitions of a mathematical function. G G Joseph, in the crest of the peacock argues that the algebraic nature of Indian mathematics arises as a consequence of the structure of the Sanskrit language. Ingerman in his paper titled Panini-Backus form finds Panini's notation to be equivalent in its power to that of Backus - inventor of the Backus Normal Form used to describe the syntax of modern computer languages. Thus Panini's work provided an example of a scientific notational model that could have propelled later mathematicians to use abstract notations in characterizing algebraic equations and presenting algebraic theorems and results in a scientific format.

Philosophical doctrines also had a profound influence on the development of mathematical concepts and formulations. Like the Upanishadic world view, space and time were considered limitless in Jain cosmology. This led to a deep interest in very large numbers and definitions of infinite numbers. Infinite numbers were created through recursive formulae, as in the Anuyoga Dwara Sutra. Jain mathematicians recognized five different types of infinities: infinite in one direction, in two directions, in area, infinite everywhere and perpetually infinite. Permutations and combinations are listed in the Bhagvati Sutras (3rd C BC) and Sathananga Sutra (2nd C BC).

Jain set theory probably arose in parallel with the Syadvada system of Jain epistemology in which reality was described in terms of pairs of truth conditions and state changes. The Anuyoga Dwara Sutra demonstrates an understanding of the law of indeces and uses it to develop the notion of logarithms. Terms like Ardh Aached , Trik Aached, and Chatur Aached are used to denote log base 2, log base 3 and log base 4 respectively. In Satkhandagama various sets are operated upon by logarithmic functions to base two, by squaring and extracting square roots, and by raising to finite or infinite powers. The operations are repeated to produce new sets. In other works the relation of the number of combinations to the coefficients occurring in the binomial expansion is noted.

Since Jain epistemology allowed for a degree of indeterminacy in describing reality, it probably helped in grappling with indeterminate equations and finding numerical approximations to irrational numbers.

Natural Language Processing is challenging area of Research areas in Artificial Intelligence[1, 2, 3, 4,, 5, 7, 8, 9, 11, 13]. Sanskrit n language processing is one of the Research field in Artificial intelligence [14, 15, 17] . Panini defined Sanskrit grammar with 4000 rules long back in sixth century B.C. In the following, The Logical representation of Panini's Grammar is discussed for Sanskrit Language. The Sanskrit Language is translated in to English character set to processes using Prolog.

. The Knowledgebase of Sanskrit is defined in English character set for logic programming using Om Transliteration[11]. The Knowledgebase Sanskrit and Tamil in English character set processed using Prolog. Examples are discussed for Sanskrit language processing.

II. LOGIC REPRESENTATION FOR SANSKRIT LANGUAGE

Beattie[2] presents an introductory review of some aspects of the computer processing of natural language in the form of string alphabetic characters, for example, spoken word. Applications of such processing in fields like information storage and retrieval and computer-assisted instruction are discussed for a computer to "understand" natural language[3].

The Sanskrit Language can be processed by defining English alphabetic characters using OM SETUP. This Sanskrit language representation in English shall be used for reasoning with the Sanskrit language

1. May be, it is.

Syadasti
The Logic representation is given by
Syadasti(x)→ Syadasti(x)
2. May be, it is not
Syad nasti
Syad nasti(x)→ ¬ Syadasti(x)
3. May be it is, and it is not at different times
Syad astinasti
Syad astinasti(x)→Syadasti(x) Λ Sada nasti(x) Λ ¬ different times (x)
Syad astinasti(x)→Syadasti(x) Λ Sada nasti(x) Λ ¬ asti(x)
4. May be it is and it is not at the same time and is indescribable
Syad avaktavya
Syad avaktavya(x)→Syadasti(x) Λ asti(x,) Λ indescribable (x)
Syad avaktavya(x)→Syadasti(x) Λ asti(x,) Λ avaktavya (x)
5. May be it is, and yet indescribable
The above sentence may be logically equaling to
May be it is, and not indescribable
Syad asti avaktavya
Syad asti avaktavya(x)→Syadasti(x) Λ ¬indescribable (x)
Syad asti avaktavya(x)→Syadasti(x) Λ ¬ avaktavya (x)
6. May be it is not, and also indescribable
Syad nasti avaktavya
Syad asti nasti avaktavya(x)→ Sada nasti(x) Λ indescribable (x)
Syad asti nasti avaktavya(x)→ ¬ Syadasti(x) Λ avaktavya (x)
7. May be it is, and it is not and also indescribable
Syad asti nasti avaktavya
Syad asti nasti avaktavya(x)→Syadasti(x) Λ Sada nasti(x)) Λ indescribable(x)
Syad asti nasti avaktavya(x)→Syadasti(x) Λ Sada nasti(x)) Λ avaktavya(x)

III. PROLOG PROGRAMMING

The Prolog is a Logic Programming language. The knowledge is represented as predicates and Clauses in Prolog. Logic programming provides simple resolution strategy The simple resolution strategy is

```
C(X, Z) :-
        A(X,Y),
            B(Y, Z)
```

It can be read as " C(X, Y) is true if A(X, Y) and B(Y,Z) is true"

The Prolog program for the above clause is

```
Predicates
A(X,Y).
B(Y,Z).
Clauses
C(X,Z):-A(X,Y), B(Y,Z).
```

The result will be given for above program for input

```
C(X,Z)
Yes
For the goal
C(X,Y)
no
```

A predicate is a relation with name of the relation and arguments . The arguments may be contain variables or constants.
for instance

```
father(x,y)
father(raama, dasaradha)
```

where father is name of the relation, x and y are variables, and raama and dasaradha are constants.

A clause is combination of and or more predicates for the rules.

For instance

```
Grandfather(X, Z) :- father(X, Y) , father (Y, Z)
```

Suppose, consider the following facts and rules

```
Raama is father of Lava
Raama is father of Kusha
Dasaradha is father of Raama
```

If X is father of Y and Y is father of Z then X is Grand father of Z

The Prolog programming may be written SWI-Prolog

```
father(kusa, raama).
father(raama, dasaradha).
father(lava, raama).
grand_father(lava, dasaradha) :- father(lava, raama), father(raama, dasaradha).
```

If the goal is given as

```
grand_father(lava, dasaradha)
```

The system will give
Yes

IV. SANSKRIT LANGUAGE GENERATION USING Om Transliteration

It is not possible to processes Sanskrit directly in Prolog because Prolog processes in English. Sanskrit language processing, First the Sanskrit language has to be represented in English character set. The Sanskrit language can be generated in English character set using Om Transliteration Editor[11].

In Om Transliteration if the sentence is given like this "lavaa pitaa raam
" equivalent English sentence "raama is father of lava"

The Sanskrit sentence is generated for "lavaa pitaa raam"
लवा पिता राम्

The Sanskrit character set shall be generated using Om Transliteration

1. May be, it is.
Syadasti
स्यदस्ति
2. May be, it is not
Syad nasti
सद नस्ति
3. May be it is, and it is not at different times
Syad astinasti
स्य अस्तिनस्ति
4. May be it is and it is not at the same time and is indescribable
Syad avaktavya
स्य अवक्तव्य
5. May be it is, and yet indescribable
Syad asti avaktavya

स्य अस्ति अवक्तव्य

6. May be it is not, and also indescribable

Syad nasti avaktavya

स्य नस्ति अवक्तव्य

7. May be it is, and it is not and also indescribable

(Syad asti nasti avaktavya

स्य अस्ति नस्ति अवक्तव्य

V. SANSKRIT LANGUAGE PROCESSING USING PROLOG

Usually Natural Language Processing will do in English. It is also possible too processes natural languages like Sanskrit by generating in English character set .

The Sanskrit Language is translated in to Logical form. The translated Logical form than represented in Prolog for Processing.

The Sanskrit Language shall be generated in English alphabetic character set using OM SETUP. This Sanskrit language representation in English shall be used for reasoning

1. May be, it is.

Syadasti

स्यदस्ति

The Logic representation is

Syadasti(x)

स्यदस्ति(x)

2. May be, it is not

Sada nasti

सद नस्ति

Sada nasti(x)→ ¬ Syadasti(x)

सद नस्ति(x)→ ¬ स्यदस्ति(x)

3. May be it is, and it is not at different times

Syad astinasti

स्य अस्तिनस्ति

Syad astinasti(x)→Syadasti(x) Λ Sada nasti(x) Λ ¬ asti(x)

स्य अस्तिनस्ति(x)→ स्यदस्ति(x) Λ सद नस्ति (x) Λ ¬अस्ति(x)

4. May be it is and it is not at the same time and is indescribable

Syad avaktavya

स्य अवक्तव्य

Syad avaktavya(x)→Syadasti(x) Λ asti(x,) Λ avaktavya (x)

स्य अवक्तव्य (x)→ सद नस्ति (x) Λ अस्ति(x) Λ अवक्तव्य (x)

5. May be it is, and yet indescribable

The above sentence may logically equal to

May be it is, and not indescribable

Syad asti avaktavya

Syad asti avaktavya(x)→Syadasti(x) Λ ¬avaktavya (x)

स्य अस्ति अवक्तव्य(x)→ सद नस्ति (x) Λ ¬अवक्तव्य(x)

6. May be it is not, and also indescribable

Syad asti nasti avaktavya

स्य अस्ति नस्ति अवक्तव्य

Syad asti nasti avaktavya (x)→ ¬ Syadasti(x) Λ avaktavya (x)

स्य अस्ति नस्ति अवक्तव्य(x)→ ¬ सद नस्ति (x) Λ अवक्तव्य(x)

7. May be it is, and it is not and also indescribable

स्य अस्ति नस्ति अवक्तव्य

Syad asti nasti avaktavya

Syad asti nasti avaktavya(x)→Syadasti(x) Λ Sada nasti(x)) Λ avaktavya(x)

स्य अस्ति नस्ति अवक्तव्य(x)→ सद नस्ति (x) Λ ¬ स्यदस्ति(x) Λ अवक्तव्य(x)

Example 1

Consider an Example for Sanskrit Language Processing with Prolog

The logical sentence can be represented in Prolog and the Sanskrit logical sentences can be generated using Om Transliteration

Consider the sentences for Sanskrit

raama is father of lava

The English character set of Sanskrit of raama is father of lava

lavaa pitaa raam

The Sanskrit sentence is generated using Om Transliteration

लवा पिता राम्

The Prolog representation is

pitaa (lavaa , raam) which is equaling to

पिता (लवा ,राम्)

Similarly it may be followed

raama is father of kusa

kusaa pitaa raam

कुसा पिता राम्

pitaa (kusaa ,raam)

पिता (कुसा ,राम्)

Dasaradha is father of Raama

raam pitaa dasaradh

राम् पिता दसरध्

pitaa (raam, dasaradh)

पिता (राम् ,दसरध्)

If raama is father of lava and dhasaradha is father of raama then dhasaradha is Grand father of lava

लवा पिता राम् राम् पिता दसरध् लवा पितामह् दसरध्

पितामह् (लवा ,दसरध्):- पिता(लवा , राम्), पिता(राम्,दसरध्)

If X is father of Y and Y is father of Z then X is Grand father of Z

X पिता Y , Y पिता Z X पितामह् Z

पितामह्(X,Z):- पिता(X,Y), पिता(Y,Z)

The Prolog programming is defined in SWI-Prolog

```
pitaa(lavaa , raam).
pitaa(kusaa ,raam).
pitaa(raam, dasaradh).
pitaamahaa(kusaa,dasaradh):-pitaa(kusaa,raam),
pitaa(raam,dasaradh).
```

If the goal is given

pitaamahaa (kusaa ,dasaradh)

The answer is given by the system

yes

If the predicate is given to Prolog

pitaamahaa (kusaa ,dasaradha)

The answer is given by the system

Yes

If the predicate is given to Prolog

pitaamahaa .(raam, dasaradh)

The answer is given by the system

no

Which is equivalent in Sanskrit Language Processing

पिता (लवा ,राम्)

पिता (कुसा ,राम्)

पिता (राम् ,दसरध्)

पितामह(कुसा, दसरध्):- पिता(कुसा, राम्), पिता(राम्, दसरध्)

If the goal is given

पितामह(कुसा, दसरध्)

The answer is given by the system

येस्

Example 2

Syadasti(x)

Syad nasti(x)→ ¬ Syadasti(x)

Syad asti nasti avaktavya (x)→ ¬ Syadasti(x) Λ avaktavya (x)

which is equalent to

Syad asti nasti avaktavya (x)→ Sada nasti(x) Λ avaktavya (x)

Syad asti nasti avaktavya(x)→Syadasti(x) Λ Sada nasti(x)) Λ avaktavya(x)

Which is equalent in Sanskrit Language Processing

स्यदस्ति(x)

सद नस्ति(x)→ ¬ स्यदस्ति(x)

स्य अस्ति नस्ति अवक्तव्य(x)→ ¬ स्यदस्ति(x) Λ अवक्तव्य(x)

Which is equalent to

स्य अस्ति नस्ति अवक्तव्य(x)→ सद नस्ति (x) Λ अवक्तव्य(x)

स्य अस्ति नस्ति अवक्तव्य(x)→ सद नस्ति (x) Λ ¬ स्यदस्ति(x) Λ अवक्तव्य(x)

Prolog Programming is defined in SWI-Prolog

```
syadasti(x).
syad_nasti(x).
avaktavya(x).
syad_nasti_avaktavya(x):-syad_nasti(x), avaktavya(x).
syad_asti_nasti_avaktavya(x):- syadasti(x),
syad_nasti_avaktavya(x).
```

If the goal is given to the system

syad_asti_nasti_avaktavya(x):

The system is given as

yes

.VI. CONCLUSION

The Natural Languages are only possible to processes with the English character set. It is possible to process other Indian languages by translating in to English character set. The main issue of Natural Language is Knowledge Representation. Om Transliteration will provide Indian Language character set into English character. The Sanskrit Grammar representation in English Character set is discussed. The Logic representation of Panini's Sanskrit Grammar is studied for Knowledge Representation. Sanskrit Language Processing using Logic Programming for Panini's Sanskrit Grammar is studied. SWI-Prolog is used for Sanskrit Language Processing. Simple examples are taken for study Sanskrit Language Processing. Once Sanskrit is translated into English character set, it is as usual processing in English.

ACKNOWLEDGMENT

The author thanks to Prof. V. V. S. Sarma, IISc., for providing literature for this work and reviewers of this paper for their voluble sugetions.

REFERENCES

[1] Allen, J.F., Natural Language Understanding, Addison Wesley, 1995

[2] J.D. Beattie, Natural language processing by computer,International Journal of Man-Machine Studies, Volume 1, Issue 3, Pages 311-329, July 1969.

[3] Gerald Gazdar and Chris Mellish. Natural Language Processing in X. Addison-Wesley.,1989.

[4] Herath, S. Ishizaki, Y. Anzai, H. Aiso, T. Ikeda , Machine processing of a natural language with interchangeable phrases, Information Sciences, Volume 66, Issues 1-2, Pages 139-165, December 1992.

[5] Grosz, Barbara J., Karen Sparck Jones, and Bonnie L. Webber, editors.Readings in Natural Language Processing. San Mateo, CA: Morgan Kaufmann. 1986.

[6] Herbert Schildt, Advanced Turbo Prolog,McGraw-Hill Inc, 1987,

[7] James Allen. Natural Language Understanding. Benjamin/Cummings,1995.

[8] James Pustejovsky, Branimir Boguraev, Lexical knowledge representation and natural language processing, Artificial Intelligence, Volume 63, Issues 1-2, Pages 193-223, October 1993.

[9] John A. Bateman, Joana Hois, Robert Ross, Thora Tenbrink,"A linguistic ontology of space for natural language processing", Artificial Intelligence, Volume

174, Issue 14, , Pages 1027-1071, Sept.2010
Keith Weiskamp and Terry Hengl, Artificial Intelligence Programming with Turbo Prolog, John Wiley & Sons Inc., 1988.

[10] Mahesh, Kavi, and Sergei Nirenburg.. Knowledge-Based Systems for Natural Language. In The Computer Science and Engineering Handbook, ed. Allen B. Tucker, Jr., 637-653. Boca Raton, FL: CRC Press, Inc.,1997.

[11] OM: one tool for many languages, Indian Institute of Science and Carnegie Milan University, Windows Version 7.1.

[12] Patterson, Dan W. Natural Language Processing. In Introduction to Artificial Intelligence and Expert Systems by Dan W. Patterson, 227-270. Englewood Cliffs, NJ: Prentice Hall.,1990

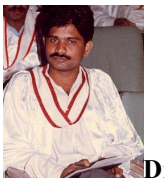
[13] Sarma, V.V.S., "A survey of Indian Logic from the point of view of Computer Science", Sadhana – Academy Proceedings in Engineering Sciences, 19,6,971-983, 1994.

[14] Sarma, V.V.S., "Computers,Sanskrit and Indian Sanskrit Traditions", Presentation at the Seminar at Kakatiya University on 25 September 2000.

[15] Subhash C. Kak The Paninian approach to natural languageprocessing, International Journal of Approximate Reasoning, Volume 1, Issue 1, Pages 117-130, January 1987.

[16] Venata Subba Reddy, P., "Fuzzy Modeling and Natural Language Processing for Sanskrit Grammar", Journal of Computer Science and Engineering, Vol.1, Issue 1, pp.99-101, may 2010.

[17] Winograd, T. Understanding Natural Language. New York: Academic Press. A pioneering work, 1972



Dr. P. Venkata Subba Reddy is Associate Professor in Department of Computer science and Engineering, College of Engineering, Sri Venkateswara University, Tirpathi, India working since 1992. He was Professor and Head, Department of Computer Science and Engineering, MeRITS, Udayagiri during 2006-07. He did M.Sc(Applied Mathematics, 1986) with Computer Programming as Specialization. He did Post Graduation Diploma in Computer Methods & Programming from Computer Society of India, Hyderabad. He did M.Phii(Database Management systems, 1988) and Ph.D(Artificial Intelligence, 1992) in Sri Venkateswara University, Tirpathi, India. . He did Post Doctoral/Visiting fellowship in Fuzzy Algorithms under Prof. V. Rajaraman, from IISC/JNCASR, Bangalore, India. He is actively engaged in Teaching and Research work to B.Tech., M.Tech., and Ph.D students. He is actively in doing research in the areas of fuzzy systems, Knowledge based systems, database systems and Natural language processing. He published papers in reputed National and International journals. He is an Editor for JCSE.