International Journal of Management and Applied Science, ISSN: 2394-7926 Volume-3, Issue-8, Aug.-2017
http://iraj.in

NLTK BASED PROCESSING OF SANSKRIT TEXT

[1]RAKHI JOON, [2]VARTIKA GUPTA, [3]CHHAVI DUA, [4]GURPREET KAUR

Department of Computer Science, Indraprastha College for Women, University of Delhi New Delhi, India
E-mail: [1]rjoon30@gmail.com, [2]vartika.gupta150198@gmail.com, [3]chhavi.gc@gmail.com, [4]chhavi.gc@gmail.com

Abstract - The paper deals with the usage and significance of Natural Language Toolkit (NLTK), an open source library for Computational Linguistics and Natural Language Processing (NLP). Extraction and processing of text is a very challenging task in the field of NLP because it requires a large corpora to be processed under various NLP tools and repositories. The research is being carried out in most of the Indian and foreign languages by analyzing the grammatical aspect of the languages and then further implementation is done using natural language repositories. In this paper the main focus is on processing Indian Languages, specifically Sanskrit that possesses a definite rule-based structure given by Panini and has a great potential in the field of semantic extraction.

Index Terms - Natural Language Processing, python, tokenizers, part-of-speech taggers, parsers, WordNet, Ashtadhayayi.

I. INTRODUCTION

India developed an extraordinarily rich linguistic tradition over more than three millennia that remained hidden for a long time. Natural languages are generally very complicated and Sanskrit is not an exception to this. Sanskrit is morphologically and lexically rich language and along with an academic subject in schools and colleges, it persists in the recitation of hymns in daily worship and ceremonies, as the medium of instruction in centers of traditional learning, as the medium of communication in selected academic and literary journals and broadcasts, and as the primary language of certain communities . The language is one of the twenty-two official languages of India in which nearly fifty thousand speakers claimed fluency in the 1991 Indian census. In the proposed work, the processing of Sanskrit text is done only with frequency measures of particular word in complete corpora. Further extension of the proposed work is also possible using other measures like tf-idf(term frequency–inverse document frequency) measures and so on. Python is chosen for implementation because it has a shallow learning curve with the syntax and semantics transparent and it has good string-handling functionality. As an interpreted language, Python facilitates interactive exploration. As an object-oriented language, Python permits data and methods to be encapsulated and re-used easily. Python comes with an extensive standard library, including tools for graphical programming and numerical processing. The paper is organized as follows: First section gives the brief overview of the research already done in the field related to NLTK especially in Sanskrit language. Second section describes NLTK, its history and advantages. In third one a Sanskrit corpus is taken via various online sources to be processed in NLTK and the result is depicted through a Lexical Dispersion plot. Conclusion and Future Scope concludes with the ongoing development and further tremendous possibilities in this area of research.

II. LITERATURE SURVEY

NLTK is used by many researchers all over the world, a few glimpse of existing work is presented in this section. In [1], the author who herself is a professor at Northern Illinois University, Department of Computer Science has described her experience of using NLP as a teaching tool to students with no previous background and the stepwise approach she developed as a result in order to acquaint students with NLTK in python. She has chosen Python specifically as it enables students to have a tiny taste of functional programming, which is supposed to improve their programming style and throughput. Similarly in [2] she described a syllabus for Introduction to NLP that concentrates on applications and motivates concepts through student experiments.

In paper [3] and [4], the author depicted the reason of selecting Python to be used for writing scripts to perform natural language processing tasks as it is a convenient language and NLTK-Lite as it provides ready access to standard corpora, along with representations for common linguistic data structures, reference implementations for many NLP tasks, and extensive documentation including tutorials and library reference. The author described the importance of Natural Language Toolkit for Computational Linguistics and also gave the idea of using NLTK as a training complex and a ready analytical tool for the development of applied text processing systems. NLTK is also integrated with Word Net, which is a database of semantic relationships for English nouns, verbs, adjectives and adverbs and also provides a detailed manual for beginners [5]. The Implementation of part of speech tagger for NLTK by using entropy method was also included in [6]. The authors described the tagging method used for replacing any of the NLTK taggers and the entropy

method used to create a part-of- speech tagger in NLTK by generating a classifier model for a set of features. The implementation methods mainly applied the feature set to the sentence and passed the resulting features to the language model. In [7], different POS taggers were developed for eight Indian Languages, namely Hindi, Bengali, Tamil, Telugu, Gujarati, Malayalam, Manipuri and Assamese which mainly involved selecting the sequences of syntactic for the words in a sentence and provided the process of writing a corpus(a way of studying natural languages, i.e deriving the set of rules for governing the languages). Along with major applications like, Text to Speech Conversion, Speech Recognition, Machine Translation etc. the tagging methods were also applied to many Indian languages and good results have been found. Since, in the paper the main focus is on Indian languages and Sanskrit language will be considered for the purpose of implementing the work in future. Only a few authors considered Sanskrit for text processing like in [8], the authors described the possible solutions to it as an efficient segmenting and tagging algorithms and dependency parsers based on constraint programming. The dependency parsers are more suitable for the analysis of Sanskrit syntax than positional grammars and constituency parsers. The authors also emphasized that unlike English, modification of the word order does not change the roles in Sanskrit and the position is highly influenced by discourse structure and emphasis. The investigation of the performance of statistical measures to determine the text-based language identification system, with an emphasis on five languages used in India based on Devanagiri script - Hindi, Sanskrit, Marathi, Nepali and Bhojpuri is done in the paper [9]. The proposed system uses n-grams as feature for classification. Language Identification is an important pre-processing step in many tasks of Natural Language Processing .The experimental details and the results obtained based on n-gram model proposed for identifying the given language pairs are also presented.

Similarly, the authors in [10] described about the dependency parser for Sanskrit that used deterministic finite automata (DFA) for morphological analysis and Utsarga Apavaada approach for relation analysis. A famous book of Sanskrit grammar 'Ashtadhayayi' was used to implement Natural Language Understanding (NLU) and Natural Language Processing (NLP) systems. To overcome the ambiguity aroused due to English grammar, a strong unambiguous grammar in form of 'Astadhayayi' by Maharishi Panini was also provided. In paper[11]the evaluation of available Part Of Speech tagsets designed for tagging Sanskrit and Indian languages which are developed in India are presented and the suitability of existing tagsets for Sanskrit from various Natural Language Processing points of view are checked.

Sanskrit presents specific challenges for computational linguistics: exact phonetic transcription in writing that obscures word boundaries, rich morphology and an enormous corpus, among others. The frequency distribution graphs were explained in [12], where the authors used the repositories provided by NLTK to carry out the processing of Hindi text and then analysis of Multi word Expressions (MWEs) was done.

III. NLTK

NLTK, the Natural Language Toolkit, is a suite of Python libraries and programs for symbolic and statistical natural language processing .NLTK includes graphical demonstrations and sample data. It is accompanied by extensive documentation, including tutorials that explain the underlying concepts behind the language processing tasks supported by the toolkit. NLTK was founded as part of a computational linguistics course in the Department of Computer and Information Science at the University of Pennsylvania by Steven Bird, Evan Klien and Edward Looper in 2001. Since then it has been developed and expanded with the help of dozens of contributors. It has now been adopted in courses in dozens of universities, and serves as the basis of many research projects. NLTK is ideally suited to students who are learning NLP or conducting research in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning. It is an open source library originally programmed in python. It is the widely used tool for solving NLP tasks and is frequently used all over the world for various text related activities. It allows students with very little previous programming experience to learn basic language processing through its user-friendly interface.

Advantages of NLTK

There are a lot of advantages of using NLTK. Because it is thoroughly documented, easy to learn, and simple to use, it has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems.

The most important advantage is that it is self-contained. Not only does it provide raw and annotated versions of real-world data in the form of 60 corpora, grammar collections, and trained models, it also provides convenient functions that can be used as building blocks for common NLP tasks. The various modules in NLTK include:

- corpus readers
- tokenizers
- stemmers

- part-of-speech taggers
- chunkers
- parsers
- classifers
- clusterers
- WordNet

NLTK defines an IDE that can be used to build NLP programs in Python. It provides basic classes for representing data relevant to natural language processing. NLTK also provides frequency and conditional frequency tools, plotting tool, and the most important thing is that everything is accessible after a single import function. All modules and corpora are provided with distributions for Windows, Mac OSX and Linux on the NLTK site.

NLTK is not only used as a training tool for research and teaching, but also as a ready analytical tool or basis for the development of applied text processing systems. Nowadays it is widely used in linguistics, artificial intelligence, machine learning projects, and so on.

IV. TEXT PROCESSING AND RESULT ANALYSIS

NLTK provides several corpora in many languages, including Indian languages under Indian Language POS-Tagged Corpus on which experiments can be performed. A new corpus can also be added into Python and can be processed using NLTK. NLTK supports various inbuilt python functions to further process the text in a particular language. In this paper, the Sanskrit corpus san.txt is obtained by extracting text from Wikipedia and other online documents and is read and processed in NLTK through the following python code -

```
>>> file = open('san.txt', encoding="utf8")
>>> text=file.read()
>>> text1=text.split()
>>> wrd=nltk.Text(text1)
>>> wrd.dispersion_plot(["ॐ","मा","ये"])
```

The positional information of ॐ, मा, ये in the entire Sanskrit text corpus is displayed using the following frequency dispersion plot [12] in *Fig.1* where each stripe represents an instance of a word and each row represents the entire text.

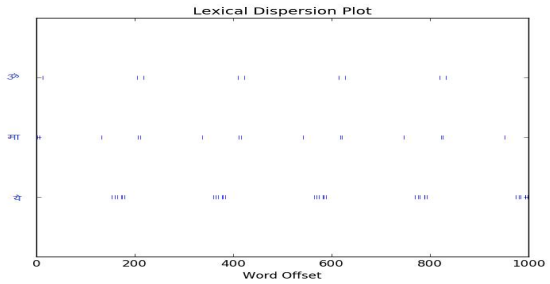

Fig 1. Frequency Distribution of three Sanskrit Words in the corpus

CONCLUSION AND FUTURE SCOPE

NLTK has undergone much development since it was released in 2001 and is still undergoing continual development as the new modules are added and existing ones are improved. Not much work is done in the Natural Language Processing of Indian languages till now, hence in this paper the work is done for Indian languages that are not popular in this area like Sanskrit. The parsing of a Sanskrit Corpus employing techniques designed is demonstrated in the paper. In this paper, only frequency distribution graphs for the processing of Sanskrit text in NLTK are included. In future, further processing and testing can also be included using NLTK for Sanskrit as well as other Indian languages which are still not very popular in research. A fully-fledged parser can be developed by improving current system. The work can be further extended using improved techniques for analyzing any Sanskrit text as well as other languages text

REFERENCES

[1] R. Freedman ,"Python as a Vehicle for Teaching Natural Language Processing ", Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2010), pp. 300-304, May 19-21, 2010
[2] R. Freedman," Teaching NLP to Computer Science Majors via Applications and Experiments", Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics (TeachCL-08), pp. 114–119, June 2008.
[3] S. Bird, "NLTK-Lite: Efficient Scripting for Natural Language Processing", Proceedings of the 4th International Conference on Natural Language Processing, pp 11-18, 2005.
[4] S. Bird," NLTK: The Natural Language Toolkit", In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, 2002.
[5] M. Romanyshyn, "Using Natural Language Toolkit for the course of Computational Linguistics".
[6] G. Malecha and Ian Smith , "Maximum Entropy Part-of- Speech Tagging in NLTK", May 14, 2010.
[7] B. Smbhavi "Current state of the ART POS TAGGING for Indian Languages – A STUDY", International Journal of Computer Engineering and Technology , pp. 250-260, May-June 2010.
[8] P. Goyal, Gérard Huet, Amba Kulkarni, Peter Scharf, Ralph Bunker, "A Distributed Platform for Sanskrit Processing ", Proceedings of COLING 2012: Technical Papers, pp. 1011–1028, December, 2012.
[9] K. Indhuja, M. Indu and C. Sreejith, "Text Based Language Identification System for Indian Languages Following Devanagiri Script", International Journal of Engineering Research & Technology, Vol. 3 Issue 4, pp. 327-331 April - 2014.
[10] S. Saxena and R. Agrawal, "Sanskrit as a Programming Language and Natural Language Processing ",Global Journal of Management and Business Studies, pp. 1135-1142.
[11] M. Gopal, D. Mishra and D. P. Singh, "Evaluating Tagsets for Sanskrit", 4th International Symposium, December 10-12 2010.
[12] R. Joon and A. Singhal, "Analysis of MWEs in Hindi Text Using NLTK", International Journal on Natural Language Computing (IJNLC) Vol. 6, No.1, February 2017
[13] Natural Language Toolkit: http://nltk.sourceforge.net
[14] NLTK Tutorial: http://nltk.sourceforge.net/index.php/Book
[15] NLTK official site: http://www.nltk.org/

★★★