COMPARISON OF NLTK AND CLTK LIBRARIES FOR PROCESSING OF PUNJABI TEXT

Kaveesh Nayak
Research Scholar
Punjabi University,
Patiala, Punjab

Abstract : ***In this paper the discussion will be related to comparison between two most common libraries used for natural language processing in python language which are CLTK and NLTK. This paper discusses the relevance of both the libraries with respect to processing of text in Punjabi language. A lot of work has been done in English language processing but work is still under development for Indian languages. The natural language processing tasks in Indian languages start with collecting and evaluating the quality corpora available in that specific language. After collecting the corpora different language specific rules are applied to the text. This paper throws light on processing a given text in Punjabi language using Python by comparing the two common libraries CLTK and NLTK.***

Index Terms – Python, Natural Language Processing, Punjabi Text, NLTK, CLTK.

I. INTRODUCTION

With the number of 22 scheduled languages in India, the scale of diversity can be imagined. Each and every language be it from India or abroad has some rules and conditions in it which make it different from other languages. One such natural language is Punjabi which is mostly used in northern parts of India (Punjab, Delhi, Himachal Pradesh, Haryana). Punjabi is a language having lots of similarity with Hindi but at the same time it also has some features totally different from Hindi which makes it unique. Punjabi is written in two scripts namely Gurumukhi and Shahmukhi. According to Wikipedia "Punjabi is 10th most widely spoken language in the world". Sri Guru Granth Sahib the holy book of Sikh religion is written in Punjabi only. Punjabi is the language used by the government in the region of Punjab (India). Not only in India, this language is also used in Pakistan though they mostly use Shahmukhi script whereas in India Gurumukhi script is used.

In this paper Punjabi Text in Gurumukhi script will be processed using two libraries in Python language and both the libraries will be compared against each other. These libraries further open way for easy implementation of statistical and probabilistic models. Python is the programming language widely accepted for Natural Language Processing. What distinguishes Python from other programming languages is that it has so many libraries which can easily facilitate all the numerical processing and graphical representation required for AI and NLP tasks. Natural Language Toolkit (NLTK) and Classic Language Toolkit (CLTK) are the libraries used for natural language processing in python. This paper will first discuss about NLTK and CLTK and in the later section of the paper comparison of both these libraries is going to be discussed with respect to text processing in Punjabi language.

II. NLTK (NATURAL LANGUAGE TOOLKIT)

The NLTK (Natural Language Toolkit) is a collection of libraries and tools in Python programming language to perform many natural language processing tasks. It was developed at University of Pennsylvania. It is so simple to use that even a person with no prior experience in coding can start learning and implementing NLP algorithms. NLTK book provided with this toolkit comes handy while learning. With NLTK come pre-installed corpora and lexical resources and includes built in tools like tokenizer and POS-tagger for some languages.

```
In [1]: import nltk
        from nltk.tokenize import word_tokenize

In [2]: text="India is a country with soul"

In [9]: nltk.word_tokenize(text)
Out[9]: ['India', 'is', 'a', 'country', 'with', 'soul']
```

Fig.1 Screenshot of word tokenizing using NLTK

III. CLTK (CLASSIC LANGUAGE TOOLKIT)

Though not as advanced as NLTK, the CLTK still holds its significance for Classic languages like English, Greek, Hindi and Punjabi. It is richer in corpora than NLTK but lacks advanced tools like POS tagger for some languages like Hindi which NLTK has.

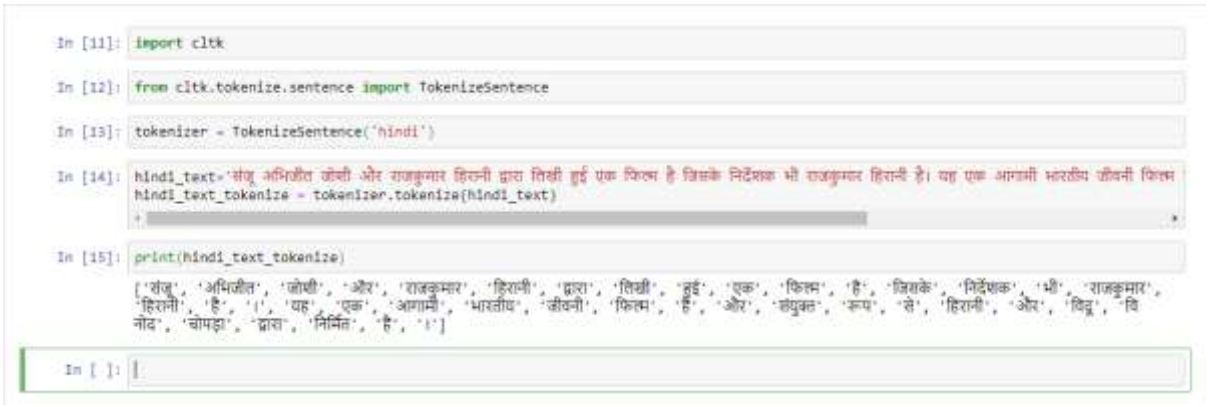

Fig.2 Screenshot of word tokenizing in Hindi using NLTK

IV. COMPARISON OF NLTK AND CLTK IN PUNJABI LANGUAGE

In this section of the paper various common functions of NLP and the corpora availability in Punjabi language will be discussed.

A. Tokenization:

Tokenization in the domain of natural language processing refers to breakage of given text into small units known as tokens. Tokens can be anything: words or numbers or symbols. The breakage of text is done with the help of some conditions known as word boundaries. Word boundaries are different for different languages.

CLTK does not have an official tokenization function for tokenization of Punjabi text. So trying the inbuilt tokenizer for Hindi works for Punjabi language also because the lexical features are almost similar for both the languages.

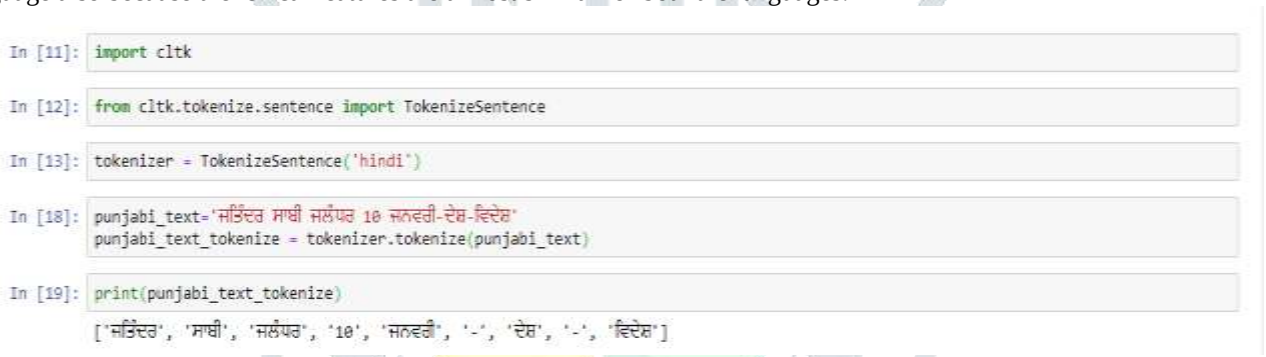

Fig.3 Screenshot of Hindi tokenizer working in CLTK on Punjabi text

NLTK also does not have a separate tokenizer for Punjabi language. But similar to CLTK this has a class IndianCorpusReader which can facilitate in tokenizing the text in Indian languages including Punjabi.

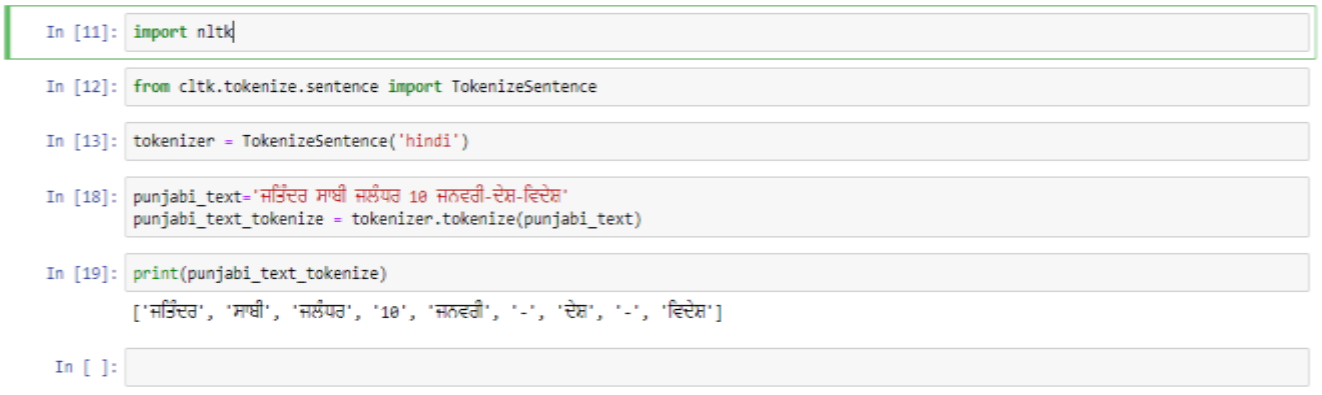

Fig.4 Screenshot of IndianCorpusReader tokenizing in NLTK on Punjabi text

B. Stop word Filtering:

Stop words are the words which are not useful for text processing in a certain language. Basically, it is a collection of words which do not contribute to the overall text in a certain language.

CLTK does have a separate stop word filtering list called STOPS_LIST which can remove stop words from text in Punjabi language.

```
In [5]: import cltk

In [14]: from cltk.stop.punjabi.stops import STOPS_LIST
         from cltk.tokenize.sentence import TokenizeSentence
         tokenizer = TokenizeSentence('hindi')

In [18]: sample = "ਨਿਆਗਰਾ ਝਰਨਾ ਅਮਰੀਕਾ ਅਤੇ ਕੈਨੇਡਾ ਨੂੰ ਪਾਣੀ ਦੀ ਲਕੀਰ ਨਾਲ ਵੱਖ ਕਰਨ ਵਾਲੀ ਇੱਕ ਰਮਣੀਕ ਥਾਂ ਹੈ ਜਿੱਥੇ ਦੁਨੀਆਂ ਭਰ ਤੋਂ ਲੋਕੀਂ ਇੱਕ ਝਲਕ ਪਾਉਣ ਆਉਂਦੇ ਹਨ।"
         sample_tokenize=tokenizer.tokenize(sample)

In [19]: print(sample_tokenize)

['ਨਿਆਗਰਾ', 'ਝਰਨਾ', 'ਅਮਰੀਕਾ', 'ਅਤੇ', 'ਕੈਨੇਡਾ', 'ਨੂੰ', 'ਪਾਣੀ', 'ਦੀ', 'ਲਕੀਰ', 'ਨਾਲ', 'ਵੱਖ', 'ਕਰਨ', 'ਵਾਲੀ', 'ਇੱਕ', 'ਰਮਣੀਕ', 'ਥਾਂ', 'ਹੈ',
'ਜਿੱਥੇ', 'ਦੁਨੀਆਂ', 'ਭਰ', 'ਤੋਂ', 'ਲੋਕੀਂ', 'ਇੱਕ', 'ਝਲਕ', 'ਪਾਉਣ', 'ਆਉਂਦੇ', 'ਹਨ', '।']

In [29]: for word in sample_tokenize:
             if word in STOPS_LIST:
                 sample_tokenize.remove(word)

In [30]: print(sample_tokenize)

['ਨਿਆਗਰਾ', 'ਝਰਨਾ', 'ਅਮਰੀਕਾ', 'ਕੈਨੇਡਾ', 'ਪਾਣੀ', 'ਲਕੀਰ', 'ਵੱਖ', 'ਇੱਕ', 'ਰਮਣੀਕ', 'ਜਿੱਥੇ', 'ਦੁਨੀਆਂ', 'ਭਰ', 'ਲੋਕੀਂ', 'ਇੱਕ', 'ਝਲਕ', 'ਪਾਉਣ',
'ਆਉਂਦੇ', '।']
```

Fig.5 Screenshot of IndianCorpusReader tokenizing in NLTK on Punjabi text

NLTK does not have a built-in list for separating stop words from a text in Punjabi language. It can only achieved by taking a large corpora and extract the words which come with maximum frequency in the corpora. So, comparing NLTK and CLTK for stop word filtering CLTK has an advantage for providing with the built in stop word list for Punjabi language.

C. Numerifier:

Numerifier is basically converting numbers written in English to numbers in Punjabi. CLTK provides built in numerifier for Punjabi language which does not exist.

```
In [5]: import cltk

In [32]: from cltk.stop.punjabi.stops import STOPS_LIST
         from cltk.tokenize.sentence import TokenizeSentence
         tokenizer = TokenizeSentence('hindi')
         from cltk.corpus.punjabi.numerifier import punToEnglish_number

In [35]: from cltk.corpus.punjabi.numerifier import englishToPun_number

In [36]: c = punToEnglish_number('੧੨੩੪੫੬੭੮੯੦')

In [37]: print(c)

1234567890

In [39]: c = englishToPun_number(1234567890)

In [40]: print(c)

੧੨੩੪੫੬੭੮੯੦

In [ ]:
```

Fig.6 Screenshot of Numerification working on Punjabi text in CLTK

D. Corpora:

Corpus is a large text in a specific language. Corpora helps in doing linguistic research in a certain language. Here again, CLTK wins because of a Punjabi corpora available already which is lacking in NLTK. The corpora available in the CLTK is the revered text of Sikh religion "Guru Granth Sahib"

E. Other functions:

Other functions like POS-tagging, normalization and stemming for Punjabi text can be performed using libraries like Scikit learn combine with NLTK and CLTK or they can also be performed using rule-based approach without any third party libraries.

V. CONCLUSION

Though work is being done in natural language processing with respect to the Punjabi language, but still a lot of work needs to be done in improving the existing techniques. Dedicated NLP libraries like NLTK, CLTK combined with the scientific libraries like Scikit learn can help in this. With Scikit learn machine learning algorithms can be applied easily to text in Punjabi language.

REFERENCES

[1] nltk.org, 'Natural Language Processing with Python'. [Online]. Available: http://www.nltk.org/book/

[2] V. Gupta, G.S. Lehal, 'Complete Pre Processing phase of Punjabi Text Extractive Summarization System', *Proceedings of COLING* 2012, pp 199-206

[3] M. Romanyshyn, 'Using Natural Language Toolkit for the course of Computational Linguistics'
[4] docs.cltk.org, 'Classic Language Toolkit'. Available: http://docs.cltk.org/en/latest/punjabi.html
[5] en.wikipedia.org, 'Natural language processing'. Available: https://en.wikipedia.org/wiki/Natural_language_processing