ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

Advancements in Machine Translation as a part of Natural Language Processing in Python

Fathimath Shouna Shayyam C A[1], Pragisha K[2]

B. Tech Student, Department of Computing Science, L B S College of Engineering, Kasaragod, India[1]

Assistant Professor, Computer Science, LBS College of Engineering, Kasaragod, India[2]

Abstract: Machine translation is the process of translating text from one natural language to other using computers. It is one of the widely researched tasks and is a sub field of Natural Language Processing (NLP). Manual translations are always time consuming and expensive. The use of machine translators enables quick and easy creation of content with a reduced manual effort. The process requires extreme intelligence and experience like a human being that a machine usually lacks. Many translators are currently available for various languages. The challenge here is in making the translator which is more efficient so that it works as spontaneously and as easily and as correctly as possible like a human. Various techniques and languages can be used to achieve this as we can see in some existing systems. But usage of python language to achieve such efficiency is discussed here with appropriate research results and examples. The day have come where we would be able to perform machine translation in python, within our favourite NLP toolkit, Natural Language Tool Kit (NLTK). The paper is based on the Python programming language together with an open source library called the Natural Language Toolkit (NLTK)

Keywords: Natural Language Processing, Machine Translation, Statistical Machine Translation, Moses, Python, NLTK, etc.

I. INTRODUCTION

The term Natural Language Processing encompasses abroad set of techniques for automated generation, manipulation and analysis of natural or human languages. Although most NLP techniques inherit largely from Linguistics and Artificial Intelligence, they are also influenced by relatively newer areas such as Machine Learning, Computational Statistics and Cognitive Science. Some very basic terminologies under NLP includes tokens, sentence, tokenization, parsing, etc. [16].

In machine translation, the goal is to have the computer translate the given text in one natural language to fluent text in another language without any human in the loop. This is one of the most difficult tasks in NLP and has been tackled in a lot of different ways over the years. Almost all MT approaches use POS tagging and parsing as preliminary steps.See below figure 2 to understand the outline of machine translation.

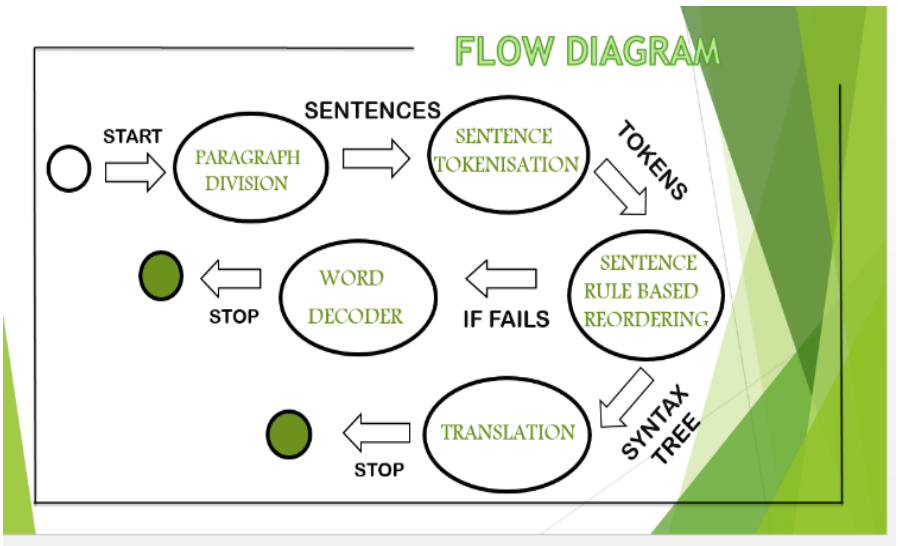


Fig 2: Machine Translation Outline

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

The Python programming language is a dynamically-typed, object-oriented interpreted language. Although, its primary strength lies in the ease with which it allows a programmer to rapidly prototype a project, its powerful and mature set of standard libraries make it a great fit for large-scale production-level software engineering projects as well. Python has a very shallow learning curve and an excellent online learning resource [11].Although Python already has most of the functionality needed to perform simple NLP tasks, it's still not powerful enough for most standard NLP tasks. This is where the Natural Language Toolkit (NLTK) comes in [12]. NLTK is a collection of modules and corpora, released under an open source license that allows students to learn and conduct research in NLP. The most important advantage of using NLTK is that it is entirely self-contained. Not only does it provide convenient functions and wrappers that can be used as building blocks for common NLP tasks, it also provides raw and pre-processed versions of standard corpora used in NLP literature and courses

II. RELATED WORKS

Many researchers, institutions and research organizations in India have started working on MT systems for English to Indian languages and among Indian languages have succeeded in obtaining very satisfactory results. The Government of India has decided to give more thrust to Language Technology for Indian languages during VIII Th plan and to initiate a program that would emphasize on quality, national relevance and participation of traditional knowledge and R & D efforts in the area of information processing in Indian languages. The Department of Electronics of Government of India launched a National level program during the year 1990-91 on Technology Development for Indian Languages (TDIL) [17]. Other institutions like IIT Kanpur, IIT Bombay, IIIT Hyderabad, University of Hyderabad, NCST Mumbai, CDAC Pune, CDAC Noida, Department of Computer Science and Engineering Jadavpur University, Kolkata, JNU New Delhi etc are playing a major role in developing the MT systems in India. Many MT systems have been developed and are being developed. The MT systems have been developed using different machine translation approaches. This paper provides brief information about development year, source & target language, translation approach, domain, salient features, and translation accuracy of major Machine Translation systems in India [18]. There is an immense need to translate these documents in respective state's local language for proper communication with common people of the state. More than 95% of the Indian population is deprived of the benefits of Information Technology due to language barrier [17].

III.EXISTING SYSTEM

Many MT systems across the globe have already been developed for the most commonly used natural languages such as English, Russian, Japanese, Chinese, Spanish, Hindi and other Indian languages etc. Figure 1 depicts the existing machine translation systems and various approaches used in developing these systems.

Table: MT approaches comparison

MT approach	Advantages	Disadvantages
Rule-based	Easy to build an initial system Based on linguistic theories Effective for core phenomena Better choice for domain specific translation The quality of translation is good for domain specific systems	Rules are formulated by experts Difficult to maintain and extend Ineffective for managerial phenomena The number of rules will grow drastically in case of general translation systems
Knowledge-based	Based on taxonomy of knowledge Contains an inference engine Interlingua representation	Hard to build knowledge hierarchy Hard to define granularity of knowledge Hard to represent knowledge
Example-based	Extracts knowledge from corpus Based on translation patterns in corpus Reduces the human cost	Similarity measure is sensitive to system Search cost is more Knowledge acquisition problem still persists
Statistical	Does not consider language grammar for translation Extracts knowledge from corpus Reduces the human errors Model is mathematically grounded	No linguistic background Search cost is expensive Hard to capture long distance phenomena Require huge amount of parallel corpora The translation quality will be very coarse due to lack of sufficient corpora

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

Automatic machine language processing was one of the first NLP application developed in computer science. Explores rule-based, statistical based, example based and knowledge based approaches. SMT is the preferred approach in many academic and industrial research. Software mainly in use here is Moses decoder

Moses: It is an open source toolkit developed it under C++ library. It is a toolkit for SMT. It is under LGPL license. It uses standard external toolkits such as GIZA++ and SRILM [14].

Statistical Machine Translation (SMT): The goal is to produce a target sentence from the source sentence that maximizes the probability. It is modelled as three separate parts:

- **Language Model (LM):** It assigns probability to any target string of words {p (e)}. An LM probability distribution over string S that attempts to reflect how frequently the string S occurs as a sentence.
- **Translation Model (TM):** It assigns probability to any pair of target and source string {p (f|e)}.
- **Decoder:** Determine translation based on probabilities of LM and TM

Limitations: Use of C++ library which is in a language which has some complexities and thus language based complexity present .Language based limitations such as error correction becomes tedious and also lengthy codes .Hence also time consuming. Also comparatively Moses are slower than NLTK

IV. PROPOSED SYSTEM

IBM Research [1] looks forward for a mission is to offer speech and language technologies that form the core of current and future products and solutions for processing natural language So this new model is with the above motive and thus overcoming the disadvantages of the existing system mostly. The system is as shown in figure 1 and with all the advancements of **python** language and **NLTK [15]**. Other software used are **NumPy**, **NLTK-data**, etc. Let us now go to each modules in the proposed system briefly.

Paragraph Division Module
The paragraph input is divided into sentences here as shown in the below figure 3.
Here we get the sentences as the output.

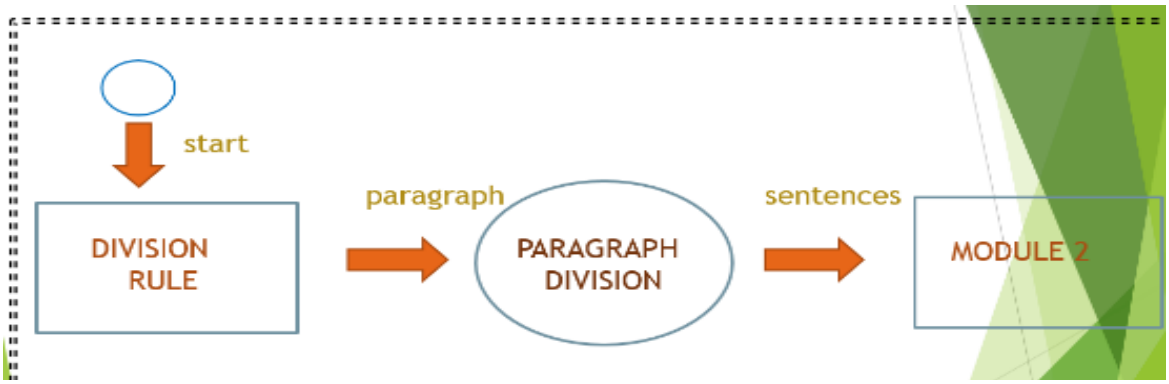


Figure 3: Paragraph Division

Sentence Tokenization Module
Sentences are divided here as tokens as shown in the figure 4.Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. Here is an example of tokenization:
Input: Friends, Romans, Countrymen, lend me your ears;

Output: | Friends | Romans | Countrymen | lend | me | your | ears |

- After tokenizing POS tagging is done in which each token is marked with its appropriate parts-of-speech tag (POS).

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

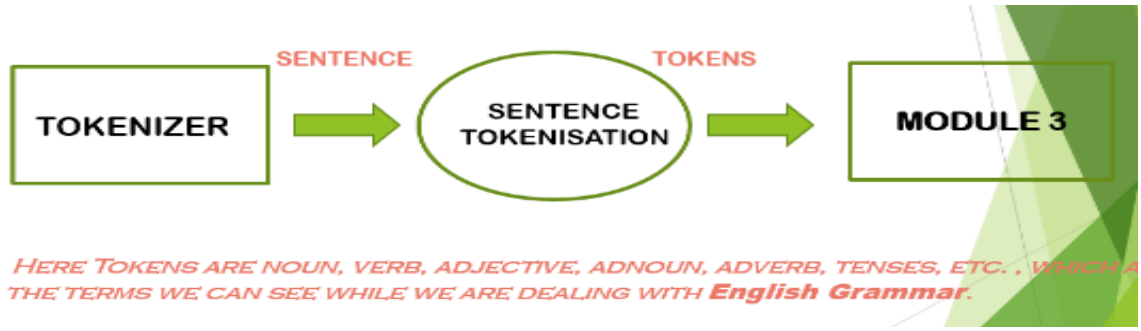


Figure 4: Tokenization

Example:

```
>>> text = word_tokenize("And now for something completely different")
>>> nltk.pos_tag(text)
[('And', 'CC'), ('now', 'RB'), ('for', 'IN'), ('something', 'NN'),
('completely', 'RB'), ('different', 'JJ')]
```

5.3 Parsing and Reordering Module

The main part under the machine translation is this module. Here the tokens are parsed into syntax trees and these trees are reordered based on the sentence grammar rule. It is as shown in the figure 5.

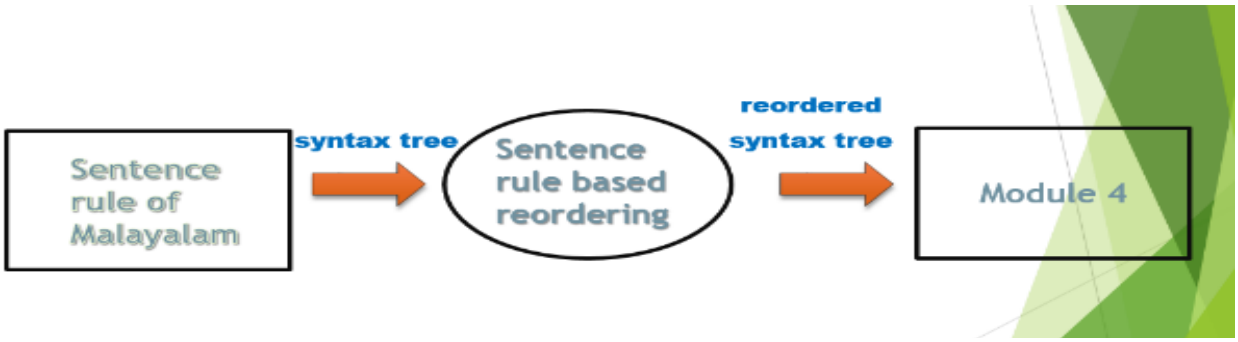


Figure 5: Parsing and Reordering

• **Parse**: Here first parse trees are created using Stanford parsers as follows
How are you dear? →

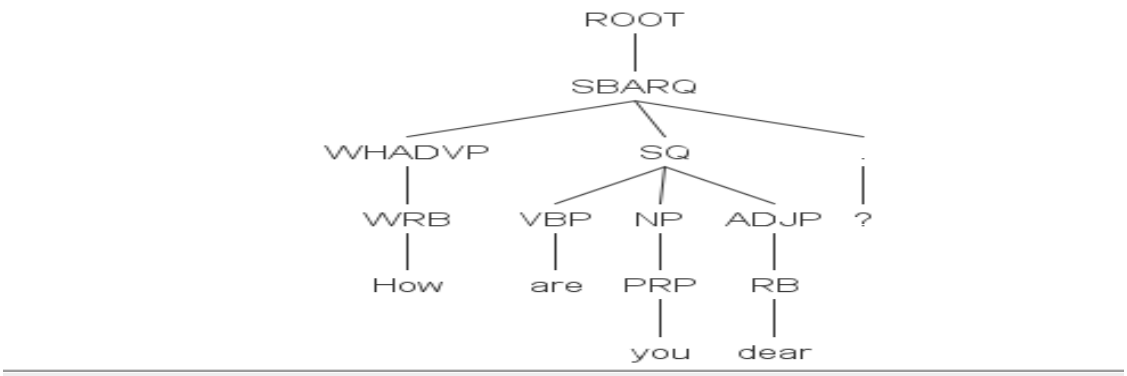


Figure: parse/syntax tree

• **Re-order:** Next we need to re order this syntax tree according to the sentence or grammar rule of the target language. For example, if the translation is from English to Tamil then

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

Production rules of English Sentence

i. S->NP VP
ii. VP->VBD NP PP
iii. PP->TO NP
iv. NP->PRP$ NN

Reordered Production rules of English sentence

i. S->NP VP
ii. VP->PP NP VBD
iii. PP->NP TO
iv. NP->NN PRP$

Example:

I. saw a beautiful child→ I a beautiful child saw
i. He came last week →He last week came
ii. Sharmi gave her book to Arthi →Sharmi her book Arthi to gave Translation Module

Here is the last step done. That is to collect all the re ordered tokens which are as the reordered tree leaves. Then replace these leaves with their corresponding meanings in target language and combine these to form a sentence again. Thus you have your text translated. It is as shown in fig6

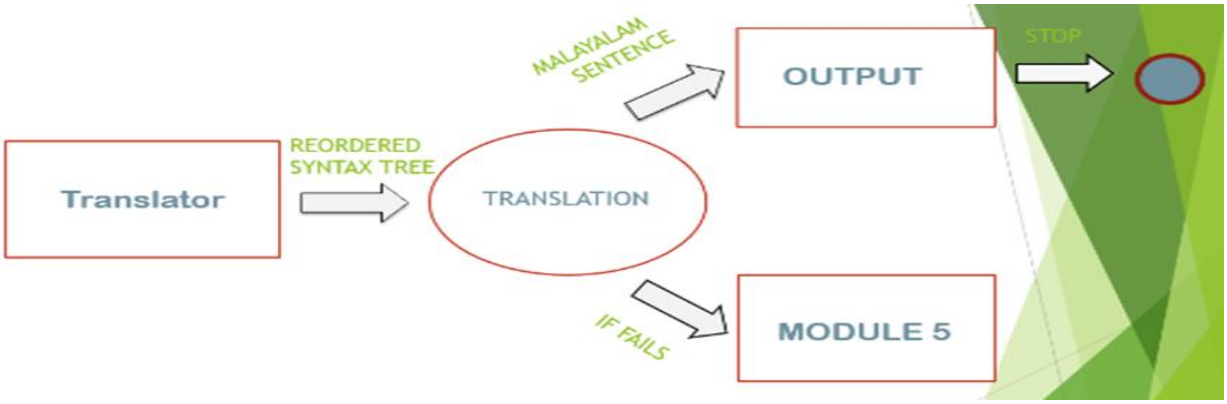


Figure 6: Translation

Example:

We can workout the system with our previous example 'Sita slept in the garden'.

Input - Sita slept in the garden

Analysis output - (S (NP (NNP Sita)) (VP (VBD slept) (PP (IN in) (NP (DT the) (NN garden)))))

After Syntactical transfer - (S (NP (NNP Sita)) (VP (PP (NP (DT the) (NN garden)) (IN in)) (VBD slept)))

Hindi lexicalization - (S (NP (NNP सीता)) (VP (PP (NP (NN बाग)) (IN में)) (VBD सोयि)))

Hindi Sentence - सीता बाग में सोयि ।

Word Decoder Module

It as shown in figure 7 it is used when the reordering module fails.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

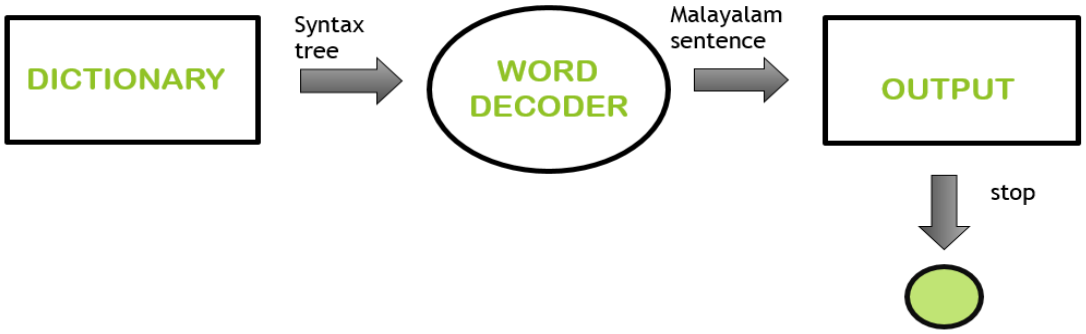
International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016



- The sentence here is not in a correct sentence rule but is the combination of each translated tokens in the input sentence, so gets a mere meaning.

Figure 7: Word Decoder

Advancements: Python is a simple yet powerful programming language with excellent functionality for processing linguistic data. Python can be downloaded for free from http://www.python .org/. Installers are available for all platforms. Python interpreters has its advantage of clearing each mistakes in codes there itself before going to next line. Supporting the NLP and MT under NLP Python provides different modules and corpora as well. NLTK is the best usage that makes the MT works so easier and relaxed NLTK is so advanced to include modules for tokenization, POS-tagging, parsing, stemming, chunking, etc. One of the most interesting advantage is the less time needed to code the projects under NLTK. Thus Python is the most advanced and suitable language with which the MT processes can be done in an efficient manner today and in future.

V. CONCLUSION

In this paper, we have described a pilot study on advancements in modelling machine translation of natural language using python. We successfully demonstrate the use of machine translation models using python and NLTK to translate code from one natural language to another.

We also demonstrated the differences in the existing MT systems with the proposed MT system and how the proposed one advances marginally. Hence, through this study we demonstrate that python programming language can be treated as a supporter of natural language processing and SMT models can be applied on them.

VI. FUTURE WORK

Some future directions can be extending these advantages of Python shown for the text translations under MT to speech translations and to other categories of NLP. Also extending these advancements in the field of Artificial Intelligence is another major theory to explore. One important avenue to explore these advancements in the mission of creating a robot as proposed by IBM in creating their dream mission **Watson, a robot who acts exactly like a human.**

ACKNOWLEDGMENT

First and foremost I wish to express my wholehearted indebtedness to God Almighty for his gracious constant care and blessings showered for the successful completion of this work. I am also thankful to Pragisha K, Professor, CSE department, LBS College of Engineering Kasaragod and other faculties and friends who had supported and directed me for the completion of this work.

REFERENCES

[1] https://www.research.ibm.com/compsci/spotlight/nlp/
[2] International Journal on Natural Language Computing (IJNLC) Vol. 2, No.4, October 2013 DOI: 10.5121/ijnlc.2013.250447 SURVEY OF MACHINE TRANSLATION SYSTEMS IN INDIA by G V Garje1 and G K Kharate2
[3] Getting Started on Natural Language Processing with Python by Nitin Madnani nmadnani@ets.org

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

[4] Introduction to Statistical Machine Translation, Phil Blunsom phil.blunsom@comlab.ox.ac.uk, Computational Linguistics Week 6, Michaelmas 2009
[5] [Stolcke, 2002] Stolcke, A. (2002). SRILM - An Extensible Language Modelling Toolkit. In Proceedings of the International Conference on Spoken Language Processing, Denver, Colorado.
[6] [Och and Ney, 2003] Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, 29(1):19–51.
[7] Statistical Machine Translation Based on Moses, Nikhil P, MCA, S4, CHINTECH
[8] Dragomir R. Radev and Kathy McKeown. 1999. Generating natural language summaries from multiple on-line sources. Computational Linguistics. 24:469-500.
[9] Adwait Ratnaparkhi 1996. A Maximum Entropy Part-Of-Speech Tagger. Proceedings of Empirical Methods on Natural Language Processing.
[10] Dekai Wu and David Chiang. 2007. Syntax and Structure in Statistical Translation. Workshop at HLT-NAACL.
[11] The Official Python Tutorial. http://docs.python.org/tut/tut.html
[12] Natural Language Toolkit. http://nltk.sourceforge.net
[13] NLTK Tutorial. http://nltk.sourceforge.net/index.php/Book
[14] Statistical Machine Translation with the Moses Toolkit Amta2014.amtaweb.org Vancouver, BC October 22-26, TUTORIAL AMT 2014 A, The 11th Conference of the Association for Machine Translation in the Americas Hieu Hoang Matthias Huck Philipp Koehn
[15] Natural Language Processing in Python .Authors: Steven Bird, Ewan Klein, Edward Loper Version: 0.9.2 (draft only, please send feedback to authors) copyright: '2001-2008 the authors License: CreativeCommonsAttribution-Noncommercial-NoDerivativeWorks3.0United States License
[16] Getting Started on Natural Language Processing with Python Nitin Madnani nmadnani@ets.org
[17] Bharati, R. Moona, P. Reddy, B. Sankar, D.M. Sharma & R. Sangal, (2003) “Machine Translation: The Shakti Approach”, Pre-Conference Tutorial, ICON-2003.
[18]Sanjay Kumar Dwivedi & Pramod Premdas Sukhadeve, (2010) “Machine Translation System in Indian Perspectives”, Journal of Computer Science 6 (10): 1082-1087, ISSN 1549-3636, © 2010 Science