European Journal of Academic Essays 1(4): 1-5, 2014
ISSN: 2183-1904
www.euroessays.org

Hindi to Bundelkhandi Language Machine Translation

<Kumar Raghvendra>[1], < Rai, Divyarth>[2,] < Jaiswal, Ashish >[3]

[1, 2,3]Dept of Computer Engineering LNCT Group of College Jabalpur, M.P., India
raghvendraagrawal7@gmail.com, divyarthrai7@gmail.com, aashish.aj12@gmail.com

Abstract: In this paper we describes a Hindi to bundelkhandi language machine translation system the developed by using a popular open source tool called Apertium tool. We are taking the example of India there are 1.20 billion people, 18 officially recognized languages that they are speaking, 30 regional languages that are the type of local language and over 2000 dialects the multi-lingual society of India needs well developed international communication tools for the Indian citizens to exchange their ideas or view to another person from another country or another district or state to put their view publically very easily. Hindi is the mother tongue for the Indian citizen that person belong to India, there are still lots of person from the different places that are unable to understand the Hindi language. If consider the area of the Madhya Pradesh there are approx 40% of people are unable to talk or even not able to understand the Hindi language. They only know their local languages that are bundelkhandi language. For that we need to help that kind of people we need to develop a tool or software that are helpful to them understand that things that, so for convert that Hindi language or English language used the Apertium platform due to several reasons. This tool is very well suited for building machine translation systems between closely related language pairs, such as Hindi to bundelkhandi due to its shallow parser level transfer modules.

Keywords: Apertium, Hindi, bundelkhandi, TAM, Anusaaraka, Transfer rules, Bilingual Dictionaries.

1. Introduction

Natural language processing is the ability to process the natural languages as much easily that understandable by the human being. So those human beings are easily able to write, speak and understand the other language. NLP [1] [2] [3]is most useful for the communication purpose between the two people when their belongings are different. And the mode of expression of the documents that they are introduces. As consider the today scenario that computer is very useful for producing the data, information, messages, spoken, written are day to day increases. So that it's very useful to the person who want to use the computer they need to understand that written language, but the computer language is mostly machine language or Hindi language. So for that need a machine translation system that is able to convert the one language to another language. Taking this type of scenario use the NLP to convert the one language into another language.

There are mainly two big challenges in the computer system that are

- *Reading and writing text or data:* - It is applied such as messaging [5] [6] [7], abstracting, writing, and entering the information in the database. With application in the big area such as artificial intelligence, office automation and libraries. Computers should be accumulate and compose the
Extended composition.

- *Translation:* - this is very important task in computer science point of view because it's very necessary to understand the spoken language or written language in the computer, person require the good translator that are able to convert the one language to another language, it's very useful for the rural are and urban area person or human.

3. Natural language processing system contains main three functions, its analyze the input and mapping that input into the meaning representation language (MRP) [7]. Reasoning about the interpretation the content to determine what should be the produced according the user input, and finally the generation of response that is different from what the user input.

2. Background

In the part we mainly focus to integrating the tense information into the machine translation. Since need both inter tense and intra tense need to analyze and extracted the tense information and after that we give the brief overview of the tense prediction.

2.1. Tense Prediction

Tense prediction [6] [7] needs to build a model based on a large numbers annotated with temporal relations and thus its focus is on how to recognize, interpret and normalize time expressions. a simple and effective data intensive approach.

Corresponding Author: Kumar Raghvendra

In particular, they trained models on main and subordinate clauses connected with some special temporal marker words, such as “after” and “before”, and employed them in temporal inference. Another typical task is crosslingual tense predication. Some languages, such as English, are inflectional, whose verbs can express tense via certain stems or suffix, while others, such as hindi often lack inflectional forms. Take hindi to bundelkhandi language translation as example, if bundelkhandi text contains particle word “(kai ka ja rau hai)”, the verb in this sentence is hai, the nearest bundelkhandi verb prefers to be translated into Hindi verb with the present tense. There are many number of authors that they are focusing or working that kind of machine translation focus on labeling the tenses for keywords in hindi language. first built a small amount of manually labeled data, which provide the tense mapping from hindi text to bundelkhandi text. Then, they trained tense based classifier to label the tense on bundelkhandi documents. There are many number of authors are proposed a parallel mapping method to automatically generate annotated data. In particular, they used hindi verbs to label tense information for bundelkhandi verbs via a parallel bundelkhandi - hindi corpus. It is reasonable to label such source side verb to client side translation process since the tense of hindi sentence is often determined by verbs. The problem is that due to the diversity of hindi verb inflection, it is difficult to map such bundelkhandi tense information into the English hindi. To our best knowledge, although above works attempt to serve for machine translation, all of them fail to address how to integrate them into a machine translation system.

2.2. Machine Translation with Tense

Dorr proposed [8] [9] the two-level knowledge representation model based on Lexical Conceptual Structures for machine translation which integrates the aspectual information and the lexical semantic information. Her system is based on an inter-lingual model and does not belong to a machine translation system. In particular, they addressed tense reconstruction on a binary taxonomy (present and past) for Chinese text and reported that incorporating lexical aspect features of telicity can obtain a 30% to 45% boost in accuracy on tense realization. It showed that incorporating latent features into tense classifiers can boost the performance. They reported the tense resolution results based on the best ranked translation text produced by machine translation system. However, they did not report the variation of translation performance after introducing tense information.

2.3. Preprocessing for Tense Modeling

In this paper, tense modeling [9] [10] is done on the target side language. Since our experiments are done on hindi to bundelkhandi machine translation, our tense models are learned only from the hindi text. In the literature, the taxonomy of hindi tenses typically includes three basic tenses present tense, past tense and future tense plus their combination with the progressive and perfective aspects. Here, we consider four basic tenses present tense, past tense, future tense and unknown tense and ignore the aspectual information. Furthermore, we assume that one sentence has only one main tense but maybe has many subordinate tenses. This section describes the preprocessing work of building tense models, which mainly involves how to extract tense sequence via tense verbs.

- *Tense Verbs*

Syntactic parse trees to find clauses [6] [9] connected with special aspect markers and collected them to train some special classifiers for temporal inference. Inspired by their work, we use parse tense tree sequence for each sentence.

Take the following three typical sentences as examples,

(a) is a simple sentence which contains two coordinate verbs, while (b) and (c) are compound sentences and (b) contains a quoted text.

(a) “Ram tum kha jar he ho”. (Present tense)

(b) “Ram tum kha gye the” (Past tense)

(c) “Ram tum kha jaoge” (Future tense)

For designing the parse tree require the root node that is able to parse the entire sentence and match the parse tree of bundelkhandi language parse tree and give the suitable required results.

Determine the tense of a node.

Input:

The Tree Node of one parse tree, leaf node;

Output:

The tense, tense;

1: Tense = Unknown
2: Obtaining the POS tag lpostag from leafnode;
3: Obtaining the word lword from leafnode;
4: If (lpostag in [ho; he: hai]) then
5: Tense = present tense
6: Else if (lpostag == the, thi, tha]) then
7: Tense = past tense
8: Else if (lpostag == jaoge]) then
9: If (word in [jaoge, jaogi, jaige]) then
10:Tense = future tense
11: Else if (lword in [the, thi, tha]) then
12: Tense = past
13: Else
14: Tense = present
15: End if
16: End if
17: Return tense;

The idea of determining the main tense is to find the tense verb located in the top level of a parse tree. According to Tree bank style, the method to determine the main tense can be described as follows:

(1) Traverse the parse tree top-down until a tree node containing more than one child is identified denoted as child node.

(2) Consider each child of target language with tag recursively traverse such node to find a tense verb. If Found, use it as the main tense and return the tense; if not, go to step (3).

(3) Consider each child of target language with tag, which actually corresponds to subordinate clause of this sentence. Starting from the first subordinate clause, apply the similar policy of step (2) to find the tense verb. If not found, search remaining subordinate clauses.

(4) If no tense verb found, return unknown as the main tense.

3. Apertium

Apertium [2] [7] is a shallow transfer type machine translation system. It basically works on dictionaries and shallow transfer rules. In operation, shallow transfer is distinguished from deep-transfer in that it does not do full syntactic parsing, the rules are typically operations on groups

of lexical units, rather than operations on parse trees. At a basic level, there are three main dictionaries:
1. The morphological dictionary for source language: this contains the rules of how words in source language are inflected. This will be called: apertium-sh-en.sh.dix
2. The morphological dictionary for target language: this contains the rules of how
words in target language are inflected. This will be called: apertium-sh-en.en.dix
3. Bilingual dictionary for the correspondences between words and symbols in the two languages. this will be called: apertium-sh-en.sh-en.dix
Writing the first source language dictionary. The dictionary is an XML file. Fire up your text editor and type the following:
<?xml version=1.0 encoding= UTF-8?>
<dictionary>
<\dictionary>
Same way create a dictionary for the target language
<?xml version=1.0 encoding= UTF-8?>
<dictionary>
<\dictionary>
For developing the bidictionary for mapping both dictionary of the source language
and the target language.
<?xml version=1.0 encoding= UTF-8?>
<dictionary>
<alphabet/>
<sdefs>
<sdef n=”n”/>
<sdef n=”sg”/>
<sdef n=”pl”/>
<sdefs>
<section id =”main” type =”standard”>
</section>
<\dictionary>
The details of each module are described below.
A de-formatter separates the text to be translated from the format information (RTF and HTML tags, white space, etc.). Format information is encapsulated so that the rest of the modules treat it as blanks between words. A morphological analyzer tokenizes the text in surface forms and delivers, for each surface form, one or more lexical forms consisting of lemma, lexical category and morphological inflection information.
1. A part-of-speech tagger which chooses, using a first-order hidden Markov model, one of the lexical forms corresponding to an ambiguous surface form.
2. A lexical transfer module reads each SL lexical form and delivers the corresponding TL lexical form by looking it up in a bilingual dictionary.
3. A structural transfer module (parallel to the lexical transfer) uses a finite-state chunker to detect patterns of lexical forms which need to be processed for word reordering, agreement, etc., and then performs these operations.
4. A morphological generator delivers a TL surface form for each TL lexical form, by suitably inflecting it.
5. A post-generator performs orthographic operations such as contractions and apostrophe photons. A re-formatter restores the format information encapsulated by the de-formatter into the translated text. Apertium also provides a utility tool to see the intermediate outputs of each.

4. Problem Definition

In this paper we are going to describe our research work for developing a machine translation system from Hindi to bundelkhandi language, where Hindi is our source language and bundelkhandi language is our target language. For developing such a machine translation system we decided to work on Apertium platform for machine transaltion. This platform provides us with the transfer engine and toolbox which allows us to build machine translation systems between any two language pairs [9]. We have to build the required data and linguistic rule to run the MT engine. We have used the following linguistic resources.
1. Morphological dictionary for Hindi language.
2. Morphological dictionary for bundelkhandi language .
3. Bilingual dictionary between Hindi and bundelkhandi language.
4. Transfer rule for Hindi to bundelkhandi language structural change.

5. Motivation

India is the largest democratic country after the china in the world with more than 30 languages and approximately 2000 dialects used for communication by the Indian people. Out of these languages English and Hindi are often used for official work. Though Hindi is recognized as national language of India, still many people exist in Madhya pradesh who neither speak nor understand Hindi. For the larger benefit of such people we need to develop an automatic Machine Translation system for various international communication tools based application. Through these not only the information exchange will be easier but a lot of knowledge sharing is also possible. Apertium is a very popular open-source platform and there
already exists many successfully built machine translation system between various European language pairs. However till date there is no such acceptable machine translation system for Hindi to bundelkhandi. So this motivated us for doing research in developing machine translation system between Hindi to bundelkhandi language.

6. Linguistic Resources

The major work that has to be done in developing such a MT system in Apertium platform is to develop the various required linguistic resources [9]. As mentioned earlier the various linguistic resources that we use in this machine translation system are
1. Morphological dictionary for Hindi language.
2. Morphological dictionary for bundelkhandi language.
3. Bilingual dictionary between Hindi and bundelkhandi language.
4. Transfer rule for Hindi to bundelkhandi structural change.
In the further sub-sections we will briefly describe the details about the file structures of this linguistic data.

A. Hindi Morphological Dictionary

It is used to get the morphological analysis of the source language i.e Hindi. The dictionary is an xml file. It uses several XML tags for writing the linguistic data. Below are the descriptions of several tags used in this dictionary file.
I. Alphabet:- It defines the set of letters that may be used in the dictionary.

II. sdef:- Defines symbols. In the context of Apertium symbol refers to a grammatical symbol label.
III. n:- for noun
IV. pl:- for plural
Other examples of symbol are
V. sg:- for singular
VI. p1:- first person
VII. pri:- present indicative
Paradigms are defined in pardef tag
VIII. e:- for entry
IX. p:- for pair
X. l:- for left and it is used for analysis.
XI. r:- for right and it is used for generation

B. Hindi- bundelkhandi Bilingual Dictionary Bilingual dictionary is also called as translation dictionary which is used to translate words or phrases from one language to another. Here we have developed a standard dictionary called shabadcose consisting of approx 30,000 words which is available as an open source dictionary between English and Hindi. We created the parallel Hindi to bundelkhandi dictionary using the words available in English-Hindi pair [1] [2].

C. Transfer rule for Hindi to bundelkhandi structural change This module is responsible for doing the structural changes from Hindi language to bundelkhandi. Normally rules are transferred from source language to target language in 3 stages known as intra-chunk, Inter-chunk and post-chunk. Intra-chunk module is responsible for doing the structural changes inside a single chunk element [9]. Inter-chunk module is responsible for doing the structural changes among the various chunks present in the input sentence and also for modifying the syntactic information associated with each chunk. Post chunk is responsible to modify the output of the inter-chunk module and to reformat it in chunk formats accepted by the generator module [1] [2] [5].

D. Generation

Final generation of the words according to their attributes is carried out by the generation module. In Apertium, the morphological analyzer and generator are the same files but differ in the direction of processing the input string. Hence the bundelkhandi morphological analyzer is used in the reverse direction for generating the final bundelkhandi translation [8][9]. By giving source language in Apertium viewer it will generate its analysis layer by layer and give its proper output. Where Apertium viewer is a utility program to view and edit output at various stages of Apertium system.

7. Conclusion and Future Work

In this paper proposed how the machine translation system from Hindi to bundelkhandi language has been developed. We described the role of Apertium tool in Hindi and bundelkhandi languages respectively. This open source model provides a great opportunity for the users to develop various applications on the top of it and improve the system as well. And Word sense disambiguation module can be attached to further tune the system. As we know that Apertium is an open source shallow transfer based machine translation system, and this work is applicable only for converting Hindi language to Bundelkhandi language, so in future we improve the output by using the output from a deep parser for handling various complex phenomena like word sense disambiguation, co-reference resolution and more.

References

[1]D. Cutting et al., “A practical part-of-speech tagger,” in Proceedings of Third Conference on Applied Natural Language Processing, Association for Computational Linguistics, Trento, Italy, pp. 133-140 ,1992.
[2] Akshar Bharati et al., “LERIL: Collaborative Effort for Creating Lexical Resources,” in Proc. of Workshop on Language Resources in Asian Languages, together with 6th NLP Pacific Rim Symposium, Tokyo, Nov. 30, 2001.
[3] Felipe Sánchez-Martínez et al., “Integrating corpus-based and rule-based approaches in an open-source machine translation system,” E-03071, Department de Lenguatges i Sistemes Informatics, Universitat d’Alacant, Alacant, Spain.
[4] Mikel L. Forcada et al., “Documentation of the Open-Source Shallow-Transfer Machine Translation Platform Apertium,” Departament de Llenguatges i Sistemes Inform`atics, Universitat d’Alacant, Alicante, Spain, Technical report, Mar. 10, 2009.
[5] Francis M. Tyers et al., "Free/open-source resources in the Apertium platform for machine translation research and development," The Prague Bulletin of Mathematical Linguistics, No. 93, pp. 67—76, 2010.
[6] Sriram Chaudhury et al., “Anusaaraka: An Expert system based MT System,” in the proceedings of IEEE conference on Natural language processing and knowledge management (IEEE-NLP KE 2010), Beijing, China, 2010.
[7] Mikel L. Forcada, “Apertium: free/open-source rule-based machine translation”, Presentation at Fourth Machine Translation Marathon “Open Source Tools for Machine Translation,” Dublin, Ireland, 29 Jan. 2009.
[8] Amba P. Kulkarni, “Design and Architecture of ‘Anusaaraka’- An Approach to Machine Translation,” Satyam Technical Review, vol 3, Oct. 2003.
[9] Akshar Bharati et al., “Natural Language Processing: A Paninian Perspective,” Prentice-Hall of India, New Delhi, 1995.
[10] Akshar Bharati et al., “Anusaaraka: Overcoming the Language Barrier in India,” appeared in “Anuvad”, Sage Publishers, New Delhi, 2002.

Author Profile

Prof. Raghvendra Kumar received B. Tech. in Computer Science and Engineering from SRM University Chennai (Tamil Nadu), India, in 2011, M. Tech. in Computer Science and Engineering from KIIT University, Bhubaneswar, (Odisha) India in 2013, and pursuing Ph.D. in Computer Science and Engineering

from Jodhpur National University, Jodhpur (Rajasthan), India. Currently he is working as Assistant Professor in Computer Science and Engineering Department at L.N.C.T Group of College Jabalpur, M.P. India. His research interests are Image Coding, Image Processing, Data Mining, NLP and Software Engineering. He has published many papers in national/International conferences and journals. He has contributed as a Technical program committee member for a number of international conferences. He is the board member of various reputed journals. He is Member of number of International association and he also published number of books in the field of computer science & engineering.