English to Devanagari Translation for UI Labels of Commercial Web based Interactive Applications

M. L. Dhore
Vishwakarma Institute of Technology,
Pune, India.

S. K. Dixit
Walchand Institute of Technology,
Solapur, India.

ABSTRACT

This paper focuses an important aspect of converting the labels on user interface of web based interactive (on-line) application in multiple languages in India. English is commonly used language over the Internet. Web based commercial on-line applications such as Internet banking usually has their user interfaces in English. The web forms of commercial web based applications has large amount of text, usually in English language text, consisting of words and phrases which appears as a part of the interaction.

This paper proposes a hybrid approach using rule based-direct Translation and human aided machine translation. Automated assistance in the generation of these text items would help to take this application typically to the rural India. In this paper, the authors describe how to create a Bi-lingual English-Marathi, English-Hindi and English-Gujarati glossary that can be used to provide automated assistance to convert the web based user interfaces. Authors have focused more on translating the labels of web based forms from English to Marathi, Hindi and Gujarati languages in India.

General Terms

Machine Translation, Natural Language Processing

Keywords

Direct Translation, Parser, Translation Memory, Bi-lingual, Multi-lingual, Transliteration

1. INTRODUCTION

Recent transformations and revolutions in information representation and retrieval represent information freely with instant access to knowledge that would have been difficult or impossible to find previously in native languages of the end user. Hence the economy and decision analysis of common users of information is based on the manipulation of information. With Internet being popular medium of information sharing, the evolution of e-Commerce and e-Governance applications recognized their acceptance largely by the society. In the case of globalization, the Internet served as a powerful and cost-effective medium of communication and information sharing with information quality preservation. As Internet is choice dependant global communication, business opportunities and sales generate vast opportunities for marketing and shipping management organizations.

While availability and usage of Internet is an indirect indicator of the growth of globalization, it has exposed the constraints placed on information sharing and readability due to linguistic barriers. The solution to this problem is Language Localization with the use of machine translation and machine transliteration. It can open the doors of the information and communication centers for those who are not conversant with the conventional language of the system. Language localization tools will be useful for bridging the gap between common people requirements and custom software. The aim of this system is to allow language access to the people by translating the labels, menus in native language [1] [2][3].

Historically, India has been a multilingual country. Today, the India Constitution officially recognizes 18 languages and ten different scripts used in different states spread across the country. An interesting point to note is that only 17.75 percent of the total population in the country can read or write in the English language [3][22]. It is evident that computer usage in India can only reach the masses if the human – computer interface is in Indian languages. Also, the success of any e-governance initiative would largely depend on the capability of the application to provide an interface to the end users in their native language. However, computer usage still seems to be a distant dream for the masses in India [4]. Hindi is the official language of India along with English and according to Ethnologue2, a well-known source for language statistics; it is the fifth most spoken language in the world. It is mainly spoken in the northern and central parts of India. Marathi is also one of the widely spoken languages in India especially in the state of Maharashtra. Both Hindi and Marathi use the “Devanagari” script and draw their vocabulary mainly from Sanskrit [5].

As the Internet continues to grow globally, Websites are providing businesses around the globe by means of viewing products & services. Over 50% of Web Users use native language other than English. Research have shown time and time again that Web users are up to four times more likely to purchase from site that communicates in user language. The Internet users can find online commercial web applications that are available in English. Few of them are On-line job applications (www.monster.com, www.naukari.com), online shopping (www.indiatimes.com), online banking (ICICI Bank), online ticket booking etc. In these applications, User Interfaces as well as End Reports generation is in English language only. Therefore, the major constraint is Linguistic Barrier, since its conception, English has been the predominant language in World Wide Web and hence its usage has been confined to the specific group of people who realize English.

An important task in localization of any interactive application in a computer system is the translation of text items that are presented by the system to the user in the course of the interaction. The text is most often in the form of words, phrases and occasionally sentences which occur in labels, commands, menus, dialogue boxes, prompt/error/help/tutorial messages, etc. In an internationalized system, such text entities are stored in separate resource files and accessed by the application through the use of a composite key consisting of the locale and the message_id. Corresponding to each locale, say English, Hindi, Marathi etc, different resource files containing the corresponding translated versions of the text entities have to be available for access by the localized applications. Creating such locale accessible resource files and compiling them for use by the application is one of the major bottlenecks faced in the localization of computer applications for users in India.

This paper describes an effort being made by the authors to develop a bi-lingual English-Hindi, English-Marathi, English-Gujarati translation memory of words / phrases occurring in the user interfaces of commercial web based on-line applications. It assists in the automatic generation of words/phrases in Hindi, Marathi and Gujarati for corresponding items in English. A very simple API has been developed so that it can be used by localization tools, which locate the English text item in the dictionary and return the corresponding Hindi, Marathi and Gujarati item either in an off-line mode to create the localized resource files or in a programmed mode providing the localized text item dynamically, as required. The process is shown in Figure.1.

2. RELATED WORK

Various machine translation (MT) systems have already been developed for most of the commonly used natural languages. In 1982, an English Japanese Machine Translation System was developed to analyze the title sentences of scientific and engineering papers by simple parsing strategies [6]. The first direct machine translation system RUSLAN is developed for closely related languages Czech and Russian [7]. The system used transfer based architecture and was rule-based. The Mantra (MAchiNe assisted TRAnslation tool) translates English text into Hindi in a specified domain of personal administration, specifically gazette notifications, office orders, office memorandums and circulars [8]. ANGLABHARTI a machine-aided translation system specifically designed for translating English to Indian languages. Anglabharti uses a pseudo-interlingua approach. It analyses English only once and creates an intermediate structure called PLIL (Pseudo Lingua for Indian Languages)[9]. Anusaaraka system translates one Indian language to another. It produces output which a reader can understand but is not exactly grammatical [10]. PONS is an experimental interlingua system exploits the structural similarity between source and target language to make the shortcuts during the translation process for automatic translation of unrestricted text [11]. CESILKO a machine translation system is localization of the texts and programs from one source language into a group of mutually related target languages [12]. Apertium uses finite-state transducers for all lexical processing operations, Hidden Markov models for part-of-speech tagging, and multi-stage finite-state based chunking for structural transfer [13]. English – Hindi translation system developed with special reference to weather narration domain [14]. VAASAANUBAADA is an Automatic machine Translation of Bilingual Bengali-Assamese News Texts using Example-Based Machine Translation technique [15]. ANGLABHARTI-II uses a generalized example-base for hybridization besides a raw example-base [16]. ANUBHARTI approach for machine aided translation is a hybridized example-based machine translation approach that is a combination of example-based, corpus-based approaches and some elementary grammatical analysis [16]. Shiva and Shakti a machine translation system has been designed to produce machine translation systems for new languages rapidly. Shakti system combines rule-based approach with statistical approach whereas Shiva is example based machine translation system. The rules are mostly linguistic in nature, and the statistical approach tries to infer or use linguistic information [17]. SisHiTr a hybrid machine translation system from Spanish to Catalan combined knowledge-based and corpus-based techniques to produce a Spanish-to-Catalan machine translation system with no semantic constraints [18]. Punjabi to Hindi Machine translation System is based on direct word-to-word translation approach as well as Hindi to Punjabi Machine translation System is also based on direct word-to-word translation approach[19][20].

3. LABEL PROCESSING SYSTEM

There are various Labels, Menus, and Dialog Boxes available on the web form of interactive web based applications. Authors have mainly focused on translating labels on the web forms of on-line banking applications. Labels can be of Single word like “Name” or can be a simple segment / phrase like “Name of the Father”. Few Labels from commercial web forms are Date of Birth or Birth Date, Account Number, Gender, Branch Code, Customer Type, Personal Details, Full Name, Name of the Father or Father Name, Name of the Mother or Mother Name, Marital Status, Residential Address, Landmark, Email Id, Signature of the Customer etc.

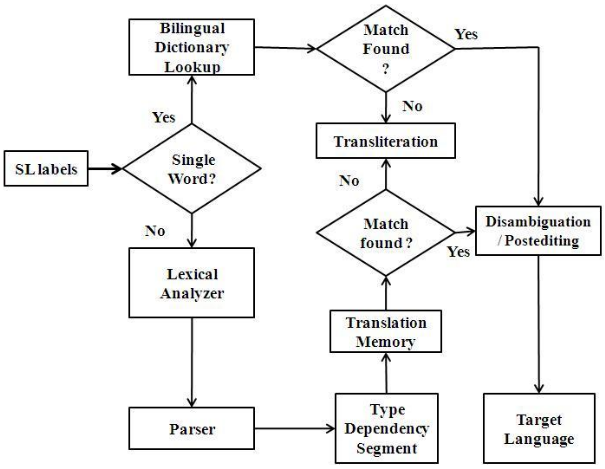


Figure 1. System Architecture

These labels have been considered as the static text on commercial web based application form. Local language user needs these labels in his native language. This can be done by using following two approaches.

I. Use the human translator to translate these labels from English to Local Language and create the Resource file of these labels which will be used by the application during the web communication.

II. Write an algorithm by using rule based method which will convert these labels from English language to local language. Direct translation with no deep analysis of syntax and semantic would be used to convert the labels offline with the little help of human interaction [23][24].

The label matching process is shown in Figure1. It would be better to use rule base direct translation using bilingual dictionary approach for domain specific applications such as banking, tourism etc. Direct translation needs analysis of source language (SL) text only as much as necessary for conversion into particular target language (TL) text. It can be done by using lexical analyzer and simple type dependency parser. Then SL words are matched with dictionary lookup followed by TL word-for-word output. TL rearrangement requires the human assistance to improve the quality of output. In rule based direct translation no deep analysis of SL syntax and semantic analysis are required. Direct translation has been proved for the domain specific applications using controlled language. It would not be proper to use Natural Language Processing to convert these labels as far as response time is considered[22][23].

3.1 TEXT PROCESSING ALGORITHM

Extract all label strings in SL from web form

```
1.  Input : Generate label_ array of label strings = { $l_1, l_2, l_3, l_4 \ldots l_n$}
2.  Input: Bilingual Dictionary = {$d_1, d_2, d_3, d_4 \ldots d_n$}
3.  Input: Translation Memory of type dependency segments = {$tm_1, tm_2, tm_3, tm4 \ldots tm_n$}
4.  repeat i=0 to label_array_length
5.  read a label $l_i$
6.  if a label $l_i$ equals to single word then
7.  repeat j=0 to bilingual dictionary length
8.  if $l_i$ equals to $d_j$ then
9.  extract all possible matches from bilingual dictionary
10. else
11. perform the transliteration
12. end if
13. end repeat
14. end if
15. if a label $l_i$ equals to multi-word then
16. create the tokens using lexical analyzer
17. generate the array of type dependency segments using parser = { $S_1$ $S_2$ $S_3 \ldots$ $S_n$ }
18. repeat m=0 to array segment length
19. read a segment $S_m$
20. repeat n=0 to translation memory length
21. if $S_n$ equals to $t_m$
22. extract all possible matches from TM
23. else
24. perform the transliteration
25. end if
26. end repeat
27. end repeat
28. end if
29. end repeat
30. disambiguate the various translation and transliteration for each label
31. replace the SL labels by TL labels
```

First it identifies all static text strings (Labels) from the web form which are written in English language and generates file of labels. For the single word label it performs the lookup in corresponding domain specific Multi-lingual dictionary shown below which is created offline by the human translators for the single independent words which occurs frequently during user interface of interactive web based applications [27]. Table 1 indicates multilingual dictionary construction formulated by implementing the algorithm.

Table 1. Multilingual Dictionary

ID	English	Marathi	Hindi	Gujarati
001	Details	माहिती, वृत्तांत विवरण,	हकीगत, हकीकत, वृत्तांत, कैफियत	વૃત્તાંત
002	Full	सगळा, अखिल, संपूर्ण, सर्व, आख्खा, अखंड	समस्त, सारा, अवघा, तमाम, पूरा, सम्पूर्ण	પુરા, સમસ્ત
003	Name	नाव	नाम	નામ
004	Account	खाते	खाता	ખાતા
005	Number	अंक, नंबर , क्रमांक	नंबर, क्रमांक	ક્રમાંક
006	Pin	पिन	पिन	પિન

3.2 LEXICAL ANALYZER

In case of multi-word label like "Name of the Father" , it uses the lexical analyzer to create the tokens according to syntactic structure of the word as follows.

```
Surname| First | Middle | Account | Number | Address | Gender | City | Country | PIN | Phone | Mobile
{ yylval=(int)strdup(yytext) ; return NOUN; }

of          { yylval=(int)strdup(yytext) ; return PREP; }

the         { yylval=(int)strdup(yytext) ; return ARTICLE; }
```

3.3 PARSER

A simple parser is used to create the type dependencies of the label segment. A parser is a program which determines if its input is syntactically valid and determines its structure. Type dependencies are automatically generated by a parser generator from descriptions of valid syntactical structures. A parser breaks data into smaller elements, according to a set of rules that describe its structure. Following is sample rule written for the parser to return the type dependency.

```
%token NOUN PREP ARTICLE
statement  : NOUN object1        {printf("%s \n",$1);return 0;}
             | NOUN object2  printf("%s %s\n",$1,$2);return 0;}
              ;
object1    : /*empty*/
             | NOUN    {printf("%s \n",$1);}
                 ;
object2 : PREP ARTICLE NOUN
            printf("%s %s %s \n",$1,$2,$3);}
```

4. RESOLVING TYPED DEPENDENCY

Typed Dependency was designed to provide simple description of grammatical relationship in a sentence, so that these could be understood and effectively used by people without linguistic expertise who wanted to extract textual relation. Each word in the sentence (except the head of the sentence) is the dependent of one other word [5].

For the label, “Name of the Student", the typed dependency will be :

_ Name of

_ of the Student

Here “of" is the preposition which decides dependency. Most of the preposition often comes before the noun form. Name is Head of the label so it cannot be dependent on the other. so “of" will depend on “Student". The process of transformation of label “Name of the Student" into Devanagari language will be as indicated in Figure 2.

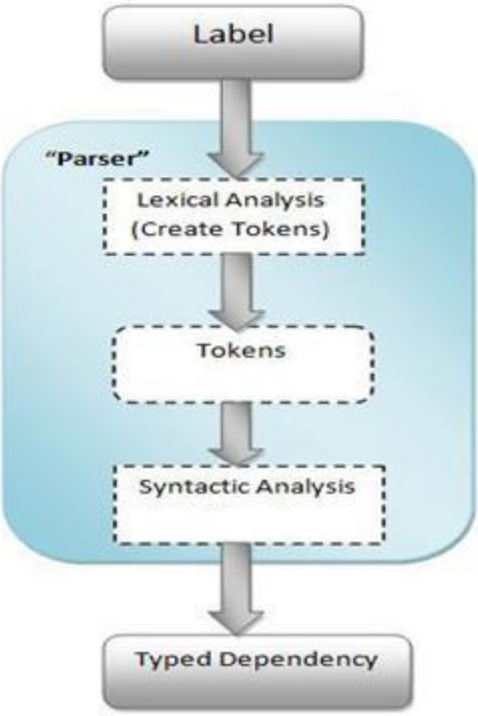


Figure 2. Process of Transformation

Figure 3 shows the real example of the transformation of the label. The values associated with the tokens are the integer values that the “flex tool” returns to the parser upon recognizing the token. We have used the Bison utility which given a context-free grammar, constructs a C program which will parse input according to the grammar rules. Then lookup for the match for the dependency segment is done with Translation memory.

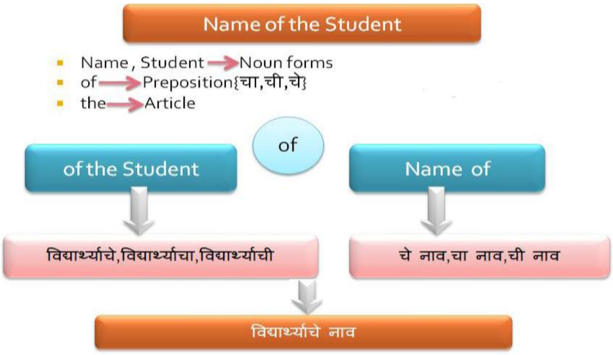

Name of the Student
Name , Student → Noun forms
of → Preposition{चा,ची,चे}
the → Article
of
of the Student
Name of
विद्यार्थ्याचे,विद्यार्थ्याचा,विद्यार्थ्याची
चे नाव,चा नाव,ची नाव
विद्यार्थ्याचे नाव

Figure 3. Example of Transformation

5. MULTILINGUAL TRANSLATION MEMORY

A translation memory is the tool that stores text segments and their translations in a database and automatically retrieves translations for text that is already in the database (usually from a previous version of the text). The tool may also find similar segments and their translations to assist the translator.

Table 2. Translation Memory of Multilingual Segments

ID	English	Marathi	Hindi	Gujarati
001	of the Father	वडिलांचे, वडिलांचा, वडिलांची	पिताजीका, पिताजीके, पिताजीकी	પિતાનુ, પિતાની, પિતાનો
002	of the Mother	आईचे, आईचा, आईची,	माताका, माताके, माताकी	માતાનુ, માતાની, માતાનો
003	personal details	वैयक्तीक माहिती , व्यक्तीगत विवरण	निजी तपशील	નિજી વૃતાંત

The idea behind TM is to store the originals and their human translations of e-Content in a computer system, broken down into manageable units. TM systems allow translators to recycle these translated segments by automatically proposing a relevant translation from the memory as a complete (exact match) or partial solution (fuzzy match) whenever the same or a similar sentence occurs again in their work [27].

Translation memory can be viewed in different forms. For specific domain, it can be developed according to subject area. Following is example of translation memory which was created for the web based commercial applications by using the type dependency parser. Table 2 shows the Translation Memory with type dependency segments generated by the dependency parser.

6. TRANSLITERATION

The words which were not found in the bi-lingual dictionary and translation memory are transliterated [3]. Multiple proper nouns of English like names of people, places and organizations, used as part of the interactive web form, are not likely to be present in the bi-lingual dictionaries. Such words are required to be transliterated to generate their equivalent spellings in the target language. Since Hindi and Marathi use Devanagari which is a phonetic script, we have used a Rule-based Transliteration approach for Devanagari to English transliteration. Since Devanagari is a phonetic script, for transliteration from Hindi/Marathi to English, we have used a rule-based approach and for transliteration from English to Hindi, we have used a segment-based transliteration approach [28][29].

In the case of text that is only partially changed (“fuzzy” matches), the translator is given the nearest matches and their translations for reference and editing. This process is called *leveraging*. The quality of results from translation memory depends on the quality of previous translations. The efficiency depends on how well the database is maintained and how proficient translators are in using the tool.

7. DIAMBIGUATION/ POST EDITING

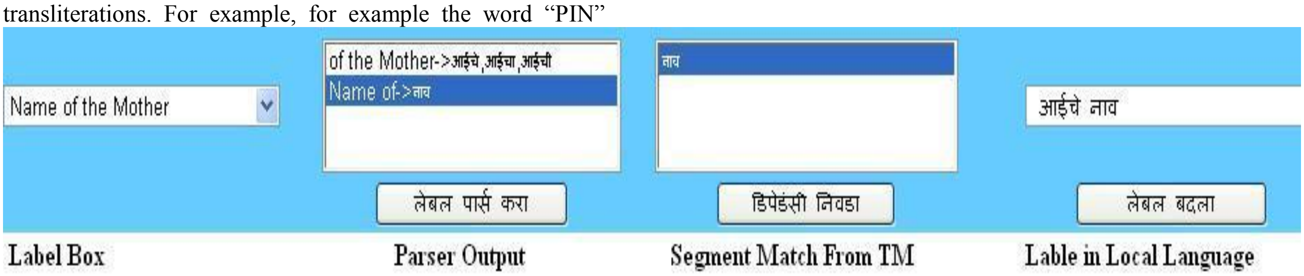
Given the various translation and transliteration choices for each word in the source language, the aim of the Translation Disambiguation module is to choose the most probable translation of the input word in the target language. In word sense disambiguation, the sense of a word is inferred based on the domain of application.

Similarly, the words in labels, although less in number, provide important clues for choosing the right translations/ transliterations. For example, for example the word “PIN” will be transliterated. Assuming we have a label with three terms, w1, w2, w3, each with different possible translations/transliterations, the most probable translation/transliteration combinations which has the maximum number of occurrences in the domain will be selected.

8. EXPERIMENTATION

We used banking glossary available on the web site of Reserve Bank of India to create the multilingual dictionary. For the lexical analyzer rules are written by using C language syntax and run on the standard tool flex to create the lexicons. For the parser, rules are written according to type dependency parser of Stanford. To generate the type dependency from the segment we used standard Bison tool for our runs. Our implementation has total four windows to support the users. First window shows all labels to be translated, second window shows all type dependency segments with the multilingual dictionary or TM match, third windows allows human expert to select the closest match and last window shows the output in target language. Figure 4 and 5 shows the snapshot of experimentation. Figure 6 shows the State Bank of India form with translated labels in Devanagari.

Performance of the system is calculated based on the quality of correct output for one word, two word, three word and four word string segments of the labels for on-line banking web based applications. Multilingual dictionary consists of 1000 words from the banking domain. The results indicate the improvements of multilingual dictionary as well as more number of semantic rules in the parser design.



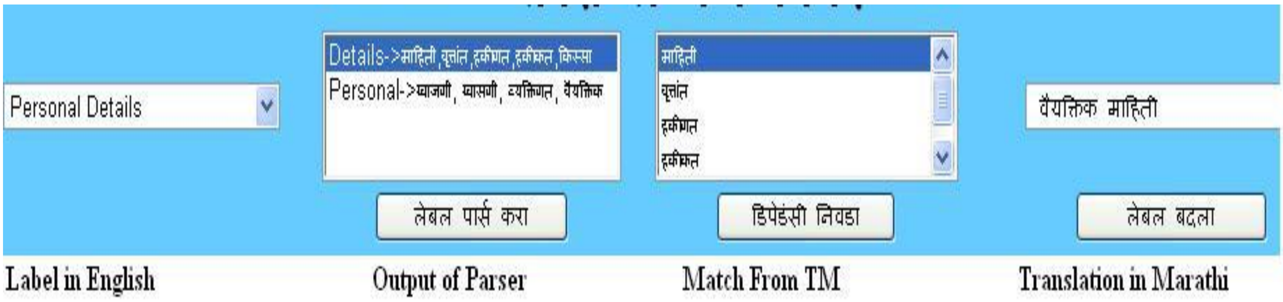
Name of the Mother

of the Mother->आईचे ,आईचा ,आईची
Name of->नाव

नाव

आईचे नाव

Figure 4. Translation of Label



Personal Details

Details->माहिती,वृत्तांत,हकीगत,हकीकत,किस्सा
Personal->ख्याजगी, ख्यासगी, व्यक्तिगत, वैयक्तिक

माहिती
वृत्तांत
हकीगत
हकीकत

वैयक्तिक माहिती

Figure 5. Translation of Label

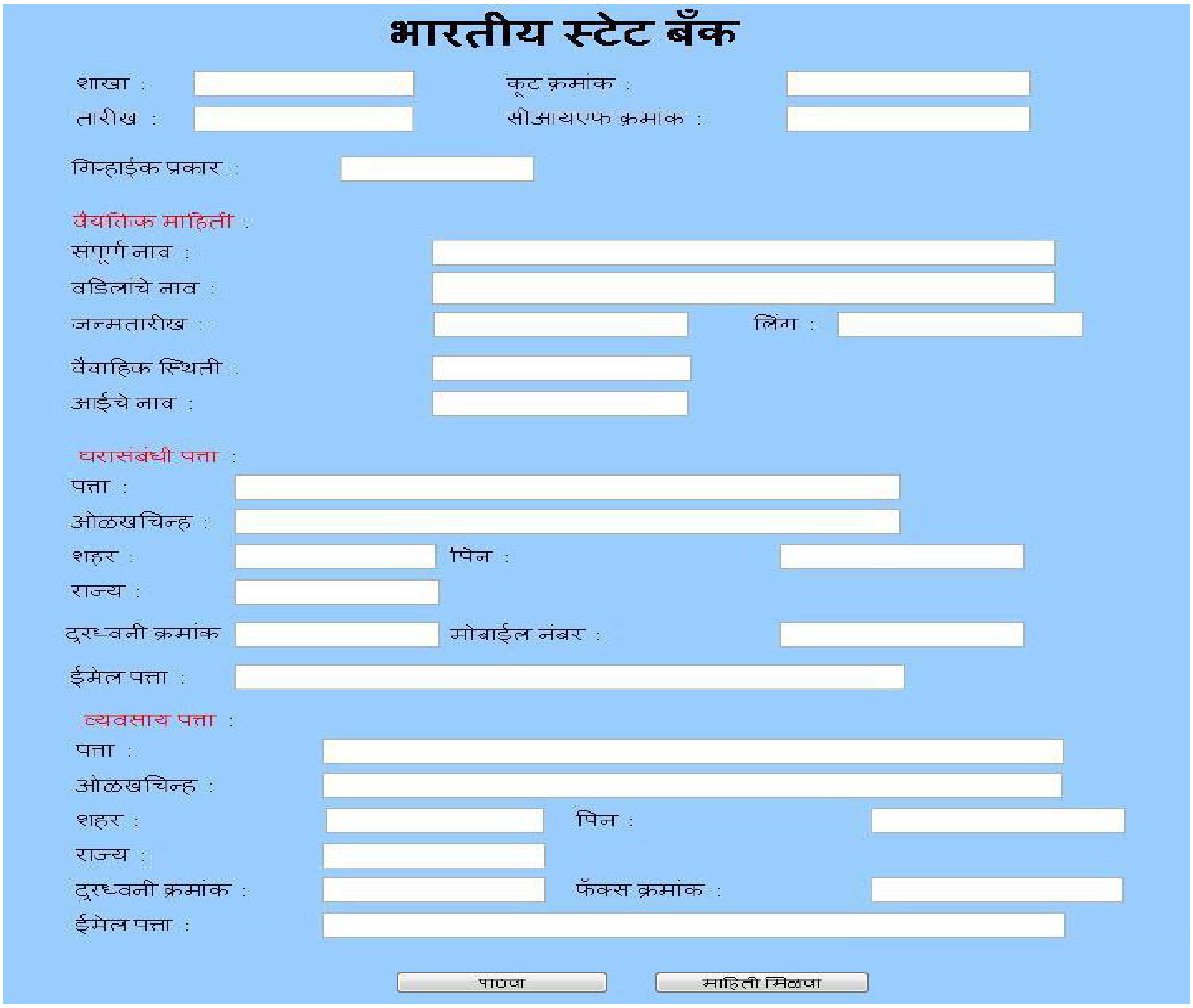


Figure 6. Translated Form of State Bank of India

Table 3. Results

SL TO TL	No of One word labels	% correct match	No of Two word labels	% correct match	No of Three word labels	% correct match	No of Four word labels	% correct match
English To Hindi	200	98.5	110	97.5	35	97	25	96
English To Marathi	200	98.5	110	97	35	97	25	95.5
English To Gujarati	200	97	110	96.5	35	96	25	96.5

9. CONCLUSION

We discussed the translation of labels of commercial web based applications from English to Devanagari (Marathi and Hindi) and Guajarati. Our approach is based on rule based direct translation and human aided machine translation. Disambiguating the various translations and transliterations is performed by providing closest synonyms for the post editing by the human expert. We have also experimented the transliterating the whole label without translation and observe that we can achieve reasonable accuracy in most of the cases. Same concept can be extended for various kinds of interactive applications, so that masses of India can comfortably use this kind of facility over the web.

10. ACKNOWLEDGMENTS

We express our gratitude to Dr. Yu. Ma. Pathan and Kishor Devdhar, authors of Sakal Papers, for Shabd Shod and Marathi puzzles, Bhalchandra Despande and Hanumant Gaikwad, Senior Librarians, Vishwakarma Institute of Technology, Pune, for providing access to the Vishwakosh Volumes, Mrs. Aparna Chandrakant Gole, family friend, Prof. Kalpana Rajendra Desai, Abasaheb Garware College, Pune, and Prof. Goraksh Garje, Pune Vidyarthi Gruha COET, Pune for their valuable guidance of Marathi linguistics. We also thank Disha, Ruchi, and Jyoti Dhore for their help in correcting the multilingual dictionary. Special thanks to Prof. A.M. Kulkarni, Prof. Umesh Kodgule and Prof. Bajirao Kondbhar who helped in the revision of this paper.

11. REFERENCES

[1] Debasri Chakrabarti,Dipak kumar Narayan, Prabhakar Pandey, Pushpak Bhattacharyya, IIT, Mumbai, “Experiences in building the Indo Wordnet- A wordnet for Hindi” , Proceeding of first global wordnet Conference

[2] Sachin Dave, Jignashu Parikh and Pushpak Bhattacharyya ,Department of Computer Science and Engineering, Indian Institute of Technology, Mumbai, India., “Interlingua-based English–Hindi Machine Translation and Language Divergence” , Journal of Machine Translation, Vol 17, 2002

[3] Katre , D.S., A Position Paper on “ Cross Cultural Usability Issues of Bilingual (Hindi & English) Mobile Phones” In the Proceedings of Indo Danish HCI Research Symposium, 2006

[4] Frost & Sullivan, “Local Language Information Technology Market in India”, TDIL, Department of IT, Ministry of Communications and Information Technology, India, 2003

[5] Nilesh Padariya, Manoj Chinnakotla, Ajay Nagesh, Om P. Damani, IIT Mumbai “Evaluation of Hindi to English, Marathi to English and English to Hindi” CLIR at FIRE 2008

[6] Makoto Nagao , Jun-ichi Tsujii , Koji Yada , Toshihiro Kakimoto, “An English Japanese machine translation system of the titles of scientific and engineering papers”, Proceedings of the 9th conference on Computational linguistics, 1982, p.245-252, July 05-10

[7] Hajic J., "Ruslan-An MT System between closely related languages", In Proceedings of the 3rd Conference of The European Chapter of the Association for Computational Linguistics, Copenhagen, Denmark, 1987, pp.113-117.

[8] MANTRA by Centre for Development of Advanced Computing, www.cdacindia.com/html/about/success/mantra.asp

[9] R. M. K. Sinha, Jain R., Jain A., “Translation from English to Indian languages: ANGLABHARTI Approach”, In proceedings of Symposium on Translation Support System STRANS,, IIT Kanpur, India. 2001

[10] Bharati, Akshar, Chaitanya, Vineet, Kulkarni, Amba P, Sangal, Rajeev Anusaaraka, “Machine Translation in stages”. Vivek, A Quarterly in Artificial Intelligence, Vol. 10, No. 3 NCST, India, 1997, pp. 22-25.

[11] Dyvik, Helge: “Exploiting Structural Similarities in Machine Translation.” Computers and the Humanities 1995, pp.225 - 234.

[12] HAJIC J, HRIC J, KUBON V., "CESILKO– an MT system for closely related languages", Tutorial Abstracts and Demonstration Notes, ACL, Washington, 2000 pp. 7-8.

[13] Carme A., Rafael C., Antonio M., Mikel L., Mireia G., Sergio O., Juan A., Gema R., Felipe S., Miriam A., “Open-source Portuguese-Spanish machine translation.”, In Lecture Notes in Computer Science 3960, 2006, pp. 50-59.

[14] Lata Gore and Nishigandha Patil, “Paper On English To Hindi - Translation System.”, Proceedings of Symposium on Translation Support Systems STRANS, IIT Kanpur, 2002, pp. 15-17

[15] Kommaluri Vijayanand, Sirajul Islam Choudhury, Pranab Ratna, “AASAANUBAADA - Automatic Machine Translation of Bilingual Bengali-Assamese News Texts.” Language Engineering Conference, Hydrabad, India 2002.

[16] R.M.K. Sinha, “An Engineering Perspective of Machine Translation: AnglaBharti-II and AnuBharti-II Architectures”, Proceedings of International Symposium on Machine Translation, NLP and Translation Support System, 2004, pp. 17-19

[17] A. Bharati, R. Moona, P. Reddy, B. Sankar, D.M. Sharma, R. Sangal, “Machine Translation: The Shakti Approach”, Pre-Conference Tutorial at ICON-2003.

[18] González J., A.L.Lagarda, J.R.Navarro, L.Eliodoro, A.Giménez, F.Casacuberta, J.M.de Val, & F.Fabregat, “SisHiTra: a Spanish-to-Catalan hybrid machine translation system”, Genoa, Italy, 2006; pp.69-73.

[19] G. S. Josan and G. S. Lehal , “A Punjabi to Hindi machine Translation System”, Coling Companion volume: Posters and Demonstrations, 2008, pp. 157-160.

[20] Sobha L, Pralayankar P,and Kavitha V., “Case Marking Pattern from Hindi to Tamil MT”, In 3rd National Conference on Recent Advances and Future Trends in IT (RAFIT),Punjabi University, Patiala, Punjab, 2009

[21] Marie-Catherine de Marneffe ,Stanford University, Linguistic Department, “The Stanford typed dependencies representation” 2008

[22] India Internet Usage Stats and Telecommunications Market Report , 2010, www.internetworldstats.com

[23] John Hutchins, University of East Anglia, Norwich , “Current commercial machine translation systems and computer-based translation tools: system types and their uses”, ourworld.compuserve.com/homepages/WJHutchins

[24] John Hutchins, University of East Anglia, Norwich , “Machine translation over fifty years”, Published in: Histoire, Epistemologie, Langage, Tome XXII, fasc. 1 2001, pp.7-31

[25] “Unicode Charts” - www.unicode.org.

[26] BarahaIME -,www.baraha.com.

[27] Edward K. Whyman And Harold L. Somers. “Evaluation Metrics for a Translation Memory System”, Software-Practice and Experience, 1999

[28] A. Ekbal and S. Bandyopadhyay, “AHidden Markov Model Based Named Entity Recognition System:Bengali and Hindi as Case Studies,” in Proceedings of 2nd International conference in Pattern Recognition and Machine Intelligence, Kolkata, India, 2007, pp. 545–552.

[29] M.L.Dhore and S.K.Dixit, “Cross Language Representation for Commercial Web Applications in Context of Indian Languages using Phonetic model”, CiiT International Journal of Artificial Intelligent Systems and Machine Learning, Vol3, No 4, April 2011, pp174-179