Lattice Based Lexical Transfer in Bengali Hindi Machine Translation Framework

Sanjay Chatterji, Praveen Sonare, Sudeshna Sarkar, and Anupam Basu

Department of Computer Sc. & Engineering, Indian Institute of Technology, Kharagpur, India
e-mail: {sanjaychatter, praveensonare007, sudeshna, anupambas}@gmail.com

Abstract

We describe a method of making proper lexical translation in Bengali to Hindi Machine Translation Framework. The baseline system replaces Bengali word with the most frequent Hindi word. But, the most frequent translation may not be the proper translation for a specific context. The proposed method finds the better lexical choice among the dictionary options with the help of the contextual information of a Hindi monolingual corpus. This approach takes Bengali sentence and converts it to Hindi sentence with the help of lattice-based data structure. The baseline and proposed translation systems are evaluated using the BLEU automatic matric and human evaluation process and later system is found performing better in both evaluations.

1 Introduction

Machine translation is a process by which a text of one(source) language is translated to a text of another(target) language. The basic unit of the machine translation is a sentence and a sentence is a sequence of words. The amount of resources is less for translation systems of majority of Indian language pairs. Since, a large amount of monolingual corpus is available in the web, they may be used for improving the quality of existing systems.

According to (Ethnologue, 2011) survey, Hindi and Bengali are the 5^{th} and 6^{th} most widely spoken languages in the world and are ranked as 1^{st} and 2^{nd} in terms of number of speakers in India, respectively. Therefore, there is a genuine requirement of translating texts of these two languages.

A synset based bilingual dictionary contains synonymous lexical items of both source and target languages. The proposed approach uses the Bengali Hindi synset based bilingual dictionary and a monolingual corpus to find the appropriate Hindi translations of the Bengali lexical items. It predicts the appropriate current Hindi lexical item based on the partial translation till the previous word of the sentence using statistical information. The system iteratively finds out the best Hindi sentence for a given Bengali sentence. One of the objectives of the proposed work is to produce desired results without using any annotated corpora.

The word lattice that is proposed in this work is a weighted directed acyclic graph with one start node and one end node. If there is an edge from node x to node y, then the word corresponding to node x is expected to appear before the word corresponding to node y. The weight of the node x and y represents the log of unigram probability of x and y respectively. The weight of the directed edge from x to y represents the log of bigram probability of (x, y).

The proposed lattice based lexical translation system has been integrated with recently released transfer based Bengali to Hindi machine translation system that has been developed under the ILMT[1] project, in place of the baseline lexical translation system.

1.1 Related Works

A few works (e.g. (Ioannou, 2003), (Lü and Zhou, 2004)) have been done related to using the monolingual corpus in the translation purpose. (Lee et al., 1999) predicted the lexical item based

[1] Indian Language Machine Translation (ILMT) is a TDIL project undertaken by the consortium of different premier institutes and sponsored by MCIT, Govt. of India.

"Proceedings of ICON-2011: 9th International Conference on Natural Language Processing, Macmillan Publishers, India. Also accessible from http://ltrc.iiit.ac.in/proceedings/ICON-2011"

on the collocation informations or syntactic relations between the words, that is calculated from dictionary suggestion and syntactically annotated monolingual corpus. Current partially correct sentence could be used to predict the next higher length partial correct sentence using a log linear monotone model with the help of score given by an interactive user interface as shown in (Barrachina et al., 2009). The bilingual synonym dictionary may help in nding parallel phrases from two monolingual corpuses (one from source language and another from target language) of same domain that will further help in statistical translation. (Haghighi et al., 2008) has done some lexical alignment from monolingual corpus.

A dynamic programming beam search algorithm for statistical machine translation has been used by Tillmann and Ney (Tillmann and Ney, 2003) for finding the best path through the lattice. We have searched the lattice monotonically using the log linear monotone model i.e., without considering the word reordering as our focus has been on the word disambiguation. Word reordering is expected to be done by the transfer grammar rules. In this scenario, monotonic search algorithm of Tillmann et al. (Tillmann et al., 1997) may also be used.

Moses (Hoang and Koehn, 2008) used the phrase based decoding for statistical machine translation (SMT). Researchers have shown the possibilities of further enhancement of phrase alignment for a particular language pair with a bilingual dictionary. Chatterji et al. (Chatterji et al., 2009) showed improvement in a Bengali Hindi SMT system by preprocessing the Moses input using dictionary. They have shown improvements in the results if the dictionary is provided as a phrase table to a phrase-based MT system like, to Moses.

Carpuat and Wu (Carpuat and Wu, 2007) showed that the lexical choices of an SMT system could be improved by the lexical choices of a Word Sense Disambiguation module. The lattice based approach described in this paper, is also applicable for the improvement of lexical selection of a Word Sense Disambiguation module and the SMT system.

The authors of (Khapra et al., 2008) have suggested an iterative approach for Hindi word disambiguation. They have used a multilingual concept based Machine Readable Dictionary for disambiguating the words of Hindi, Marathi and English. They have not tested the quality of Bengali word disambiguation and have not shown the improvement of the quality of the machine translation output with the help of their approach.

We have used SRILM (Stolcke, 2002) language model tool for estimating the probability values. SRILM calculates n-gram probabilities and takes the log of those probabilities with base 10.

2 Proposed Work

In Machine Translation, one of the most important tasks is to translate one source language word into an appropriate target language word. The baseline replacement scheme replaces the source language word by the target language word which is most frequent in terms of its individual occurrence. The proposed approach selects the word, from the Hindi word choices, that appear more frequently in a corpus in terms of individual occurrences and contextual information.

The Bengali Hindi SMT system gives an initial Hindi translation. This translation can be improved by revisiting the words and selecting the word amongst the synonymous Hindi words, that appear more frequently in a corpus in terms of individual occurrences and contextual information. This selection at each position may be done by the proposed statistical approach.

50 Bengali sentences are passed through the Bengali Hindi transfer based system. On this test set, it is found that the proposed lexical transfer method improves the BLEU score over the baseline replacement method.

2.1 System Architecture

The architecture of the proposed scheme is shown in Figure 1. As a preprocessing stage we create the language model file with the help of 500K sentence Hindi monolingual corpus where each word is stemmed into its lemma. We use the SRILM toolkit to calculate the log (base 10) of the unigram and bigram probability values. Now, the input Bengali sentence is analyzed. A Bengali Hindi bilingual synonym set dictionary, containing 20,000 concepts, is used by the *find synonym* module to find the list of synonyms that can replace the Bengali words. The *lattice building* module takes this synonym list and the log probability values to build the lattice. The *finding the best path* module uses the lattice to find the best Hindi lexical items at each level of the lattice. Fi-

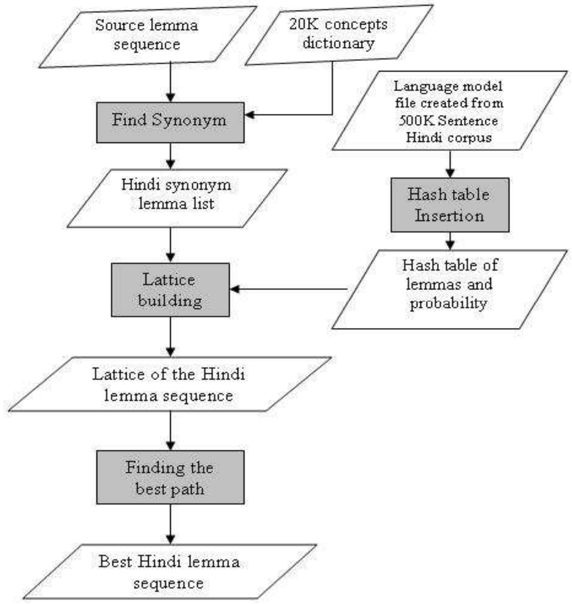

Figure 1: System Architecture

nally, the Hindi lexical sequence is used to generate the Hindi sentence. Examples are given in Itrans format. Mapping of Bengali and Hindi Unicode with Itrans is given in Appendix A.

2.2 Description of Proposed Approach

2.2.1 Word Lattice:

The lattice we have used in this context represents one source sentence and has a fixed number of levels which is the number of source sentence words. The nodes in each level correspond to the Hindi synonym words. Thus, all the paths from start to end node have same length (level). A word lattice can represent an exponential number of target sentences. However, for the computational simplicity we have considered a maximum of 10 words possible in each level of the lattice.

level = # words in source sentence = # words in target sentence

nodes in a level = # synonymous target lexicons for a source lexicon ≤ 10

2.2.2 Best Hypothesis Searching in Lattice:

We have put the unigram values as node weight and bigram values as edge weight used them in dynamic programming based beam search algorithm for searching the best path through the lattice. The recursive method for searching the best hypothesis from the lattice using the beam search algorithm, as suggested by (Tillmann and Ney, 2003), is described below. We have simplified the recursive method according to our requirement and used it to find the best hypothesis.

Let us assume that the source lexical sequence is $f_1, f_2, \ldots, f_J$. C is a coverage set of source sentence positions with increasing cardinality. Furthermore, assume that, j and j' are the last covered positions and e and e' are the last two target words. If the last partial hypotheses are $Q''(e', C - \{j\}, j')$, then the probability of best partial hypothesis Q' follow the following recursive equation.

$$Q'(e, C, j) = p(f_j|e) \times \underset{\substack{e' \\ j' \in C - \{j\}}}{Max} \{p(j|j', J)\, p(e|e')\, Q''(e', C - \{j\}, j')\}$$

In our work, all the target words are synonyms and they can all be replaced by the source word f_j. So, we are considering the translation probability $p(f_j|e)$ as 1. In the lattice structure, we have implemented in our work, the number of levels of the lattice is same as the number of words in the source sentence. We are traversing the lattice in increasing cardinality of source sentence position as well as target sentence position. Therefore, the distortion of $p(j|j', J)$ is also made 1. So, the best partial hypothesis Q' can be rewritten as follows .

$$Q'(e, C, j) = \underset{\substack{e' \\ j' \in C - \{j\}}}{Max} \{p(e|e') Q''(e', C - \{j\}, j')\}$$

The best hypothesis is selected recursively for all e which come after e' i.e., $p(e|e') \neq 0$. However, in the proposed approach, we want to consider only some of the current words that are synonym of each other and that are suggested by the bilingual synset dictionary (W). In order to achieve this, the best hypothesis for each synonymous word is multiplied by its unigram probability. The words which are not in the synset dictionary are multiplied by 0. Finally, the best partial hypothesis is selected using the following equation.

$$Q'(e, C, j) = \begin{cases} p(e) \underset{\substack{e' \\ j' \in C - \{j\}}}{Max} \{p(e|e') Q''\} & ; \quad \text{If } e \in W \\ 0 & ; \quad \text{Otherwise} \end{cases}$$

$$\left\{ where, Q'' = Q''(e', C - \{j\}, j') \right.$$

Taking log of the probability values (logp), we select the best hypothesis using the equation given in equation 1.

$$Q^{'}(e,C,j) = \begin{cases} L + \underset{\substack{e^{'} \\ j^{'} \in C-\{j\}}}{Max} \{L^{'} + Q^{''}\} & ; \quad \text{If } e \in W \\ -\infty & ; \quad \text{Otherwise} \end{cases} \tag{1}$$

$$\begin{cases} where, Q^{''} = Q^{''}(e^{'}, C-\{j\}, j^{'}) \\ and, L = logp(e); L^{'} = logp(e|e^{'}) \end{cases}$$

2.2.3 Word Lattice Searching:

To select the best target lexical sequence for a source lexical sequence, we apply equation 1. The algorithm 1 implements that equation to calculate the best partial hypothesis for the current synonym list (W) and for the list of best hypotheses of all partial sentences (S) traversed till the previous level of the lattice structure. $S^{'}$ is the list of best hypotheses of all partial sentences that are traversed till the current level. In the algorithm, we get the value of $logp(e)$ and $logp(e|e^{'})$ of the current synonym word $W[j]$ from node weight $(wg(W[j]))$ and edge weight $(wg(S[i], W[j]))$ of the lattice, respectively. The best hypothesis of each $S[i]$ is stored separately as $Q^{''}(S[i])$ function. This algorithm has time complexity of $O(length[W] \times length[S])$ and is executed for each member of the source lexical sequence($\overline{S}$). Thus, the total time complexity for finding the best path (beam) through the lattice is $O(length[W] \times length[S] \times length[\overline{S}])$

Algorithm 1 Lattice Search

Require: Partial Hypothesis $Q^{'}(S)$ and Current Synonym List W
  if $length[W] = 0$ **then**
    Return optimal word sequence.
  end if
  $Q^{''} = Q^{'}$
  for $j \leftarrow 1$ to $length[W]$ **do**
    $key = Q^{''}(S[1]) + wg(S[1], W[j])$
    for $i \leftarrow 2$ to $length[S]$ **do**
      if $Q^{''}(S[i]) + wg(S[i], W[j]) > key$ **then**
        $key = Q^{''}(S[i]) + wg(S[i], W[j]$
      end if
    end for
    $S^{'} \leftarrow key + wg(W[j])$
  end for
  Recursive Call (Lattice Search $Q^{'}(S^{'})$, next synonym list)

2.2.4 Reordering Instance:

In Bengali to Hindi translation, the reordering of words is rare though there are some cases where we find the reordering.

3. *oi ba;i duTi AmAra.* $\rightarrow$ *baha kitAba do hamArA hai.*
   that book two mine
   (Those two books are mine.)

In Bengali sentences, "duTi" quantifier can come after the noun being quantified. On the other hand, in Hindi sentences, quantifier always come before the noun. So, it may be required sometime to reorder the words in Hindi. Nonmonotonic search takes care of all such reorderings. In the proposed system, instead of nonmonotonic search we have reordered some specific instances (such as the noun followed by quantifier) in a preprocessing stage and used the monotonic search through the lattice.

2.3 Data Preparation

2.3.1 Bilingual Concept Dictionary:

In this work, a Bengali Hindi bilingual dictionary that contains 20,000 concepts is being used. Each concept contains a set of synonymous root words (lemma), each of which can be used in a specific instance. The dictionary is built as a part of the ILMT project.

2.3.2 Stemmed Monolingual Hindi Corpus:

We have crawled a 500K sentence tourism corpus from the web. Instead of annotating the corpus manually, each word of the corpus is first stemmed into its lemma. SRILM tool uses this stemmed corpus to calculate the n-gram probability values.

3 Experimental Results and Discussion

3.1 Discussion with an Example Sentence

Bengali Input: *to sAhasa kare tairI hana prAYa 24 ghanTAra eka AnandamaYa jArnIra janya .*
English: *So, get ready for an almost 24 hour's joyful journey.*

The lemma and its POS category for each Bengali word is shown in Table 2. Hindi synonym list for each Bengali lemma is given below.

to (to) sAhasa (sAhasa, himmata, dilerI, bahAdurI, majAla, hausalA, jurrata)
karA (karanA, banAnA, kara_denA, banA_denA, kArya_karanA, karAnA, karavAnA)
tairI (banAnA, taiYAra, ba.NdhAI, bA.NdhanA)
haoYA (honA, ho_jAnA, AnA, Thika_honA,

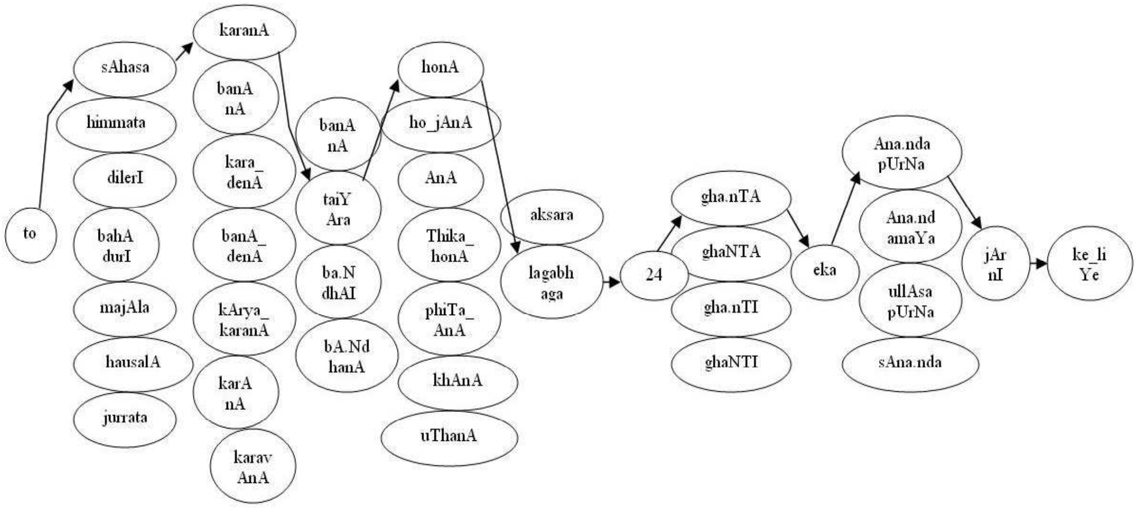

Figure 2: Lattice for the example sentence

Node	Weight	Node	Weight	Edge	Weight
to	-4.034250	honA	-4.424004	sAhasa karanA	-3.383014
sAhasa	-4.034250	AnA	-3.253551	bahAdurI karanA	-4.639096
himmata	-4.336805	khAnA	-3.253551	banAnA AnA	-5.110318
bahAdurI	-4.336805	aksara	-4.034250	taiYAra honA	-5.145906
karanA	-3.253551	lagabhaga	-3.125301	lagabhaga 24	-3.220525
karavAnA	-5.067657	24	-3.775485	24 gha.nTA	-3.105245
banAnA	-4.424005	gha.nTA	-3.614448		
taiYAra	-5.067657	eka	-4.765987		

Table 1: Weights of nodes and edges of the lattice that is shown in Figure 2

phiTa_AnA, khAnA, uThanA)
prAYa (aksara, lagabhaga) 24 (24) ghanTA (gha.nTA, ghaNTA, gha.nTI, ghaNTI)
eka (eka) AnandamaYa (Ana.ndapUrNa, Ana.ndamaYa, ullAsapUrNa, sAnanda)
jArnI (jArnI) janya (ke_liYe)

The lattice for the sentence under consideration is given in Figure 2. The weights of nodes and edges in terms of log probabilities are given in Table 1. The nodes are shown by the words and the directed edges from one node to another node are shown by two words in the table. The nodes and edges which are not listed in the table have not appeared in the corpus. Applying Algorithm 1 on the lattice the best Hindi lemma sequence, as shown in Figure 2, is given below.

Hindi Lexical Sequence: *to sAhasa karanA taiYAra honA lagabhaga 24 gha.nTA eka Ana.ndapUrNa jArnI ke_liYe .*

Translation using baseline system: *to sAhasa karake banA ho yaiye aksara 24 gha.nTA eka Ana.ndapUrNa jArnI ke liYe .*

Translation using proposed system: *to sAhasa karake* ***taiYAra*** *ho yaiye* ***lagabhaga*** *24 gha.nTA eka Ana.ndapUrNa jArnI ke liYe .*

The most frequent Hindi translations of the Bengali lemmas “tairI” and “prAYa” are “banAnA” and “aksara” respectively. But, the proposed approach replaces them to Hindi lemmas “taiYAra” and “lagabhaga”. These translations are correct with respect to the context of the current sentence.

Some of the examples of Bengali to Hindi translation through baseline and proposed system are shown in Table 3. Here, the first line of each row is the input Bengali sentence, second is the Hindi translation through baseline system, third is the Hindi translation through the proposed system and fourth line is the English translation. The

Word	lemma	POS	Word	lemma	POS	Word	lemma	POS
to	to	particle	hana	haoYA	verb	ghanTAra	ghanTAra	noun
sAhasa	sAhasa	noun	prAYa	prAYa	adverb	AnandamaYa	AnandamaYa	adjective
kare	karA	verb	24	24	number	jArnIra	jArnI	noun
tairI	tairI	noun	eka	eka	number	janya	janya	postposition

Table 2: Lemma and Parts-Of-Speech of each Bengali word.

I: ei jAtIYa phula ekhAne prachura phoTe kintu sakAle nA gele sheSha haYe yete pAre . B: ye jAtIya phUla yahA.N bahuta khilatA hai kintu subaha na yAne se a.Ntima ho sakatA hai . P: **yaha** **kisma-ke** phUla yahA.N kAphI khilatA hai lekina subaha na yAne se **khatama** ho sakatA hai . E: This kind of flowers blossom largely here but they may be finished if you don't go in the morning.
I: ei dbIpagulora madhye nau yAtAYAta chAlu chhila ki nA ? B: ye dbIpo.N ke bIcha nau yAtAyAta chalatA thA kyA nahI.N ? P: **yaha** dbIpo.N ke bIcha nau yAtAyAta jArI thA **iyA** nahI.N ? E: Between these islands boat transportation were used or not ?
I: mAsakhAneka Age theke shuru karo kArana khelATi besha bipajjanaka . B: eka mahinA Age se shuru karo kArana ye khela ThIka jokhimapUrNa hai . P: eka mahinA **pahale** se shuru karo kArana **yaha** khela **kAphI** jokhimapUrNa hai . E: Start one month before schedule because the game is quite dangerous.
I: yakhana tApamAtrA kame yAYa takhana jAYagA pAoYA besha kaThina . B: yaba tApamAna kama yAtA hai taba jagaha milanA ThIka kaThina hai . P: yaba tApamAna **ghaTa** yAtA hai taba jagaha milanA **kAphI** musakila hai . E: When the temperature decreases, its very difficult to get place.

Table 3: Translation of Bengali sentences through Baseline and the Proposed systems: I = Bengali Input, B = Baseline System, P = Proposed System, E = English.

lexical translations which are different in baseline and proposed systems are underlined. When the proposed system selects the better word then that is made bold. However, in other cases, both the translations are equally applicable.

3.2 BLEU Results

The Bengali Hindi transfer based machine translation system include lexical transfer module, that translates a Bengali word into the Hindi word which is most frequent in terms of individual accorrence. Replacing the existing lexical translation module of the baseline system by the proposed lattice based lexical translation module, we have prepared a modified system.

We have collected 50 tourism sentences which consist of 9 to 18 words each. The choice of the Hindi word corresponding to a Bengali word using the baseline MT system and the modified one may not match though both results are correct. So, we have built two reference outputs; one for the baseline and another for the modified system. The translation of those 50 sentences using the baseline and the modified systems are evaluated using the BLEU autometic metric. BLEU score of the baseline system is found to be 0.2591, whereas the corresponding score of the modified system is 0.2945. Thus, a significant improvement in BLEU score is observed.

3.3 Time Efficiency

The approach described in this paper is based on a large sized Hindi corpus crawled from web. A common difficulty in accessing a large corpus is due to its time and space complexities. In such a scenario, a hash table bashed implementation is expected to meet the expectation. We have also reduced the time to find the best path (beam) through the lattice that is constructed using unigram and bigram probability values. Instead of multiplying the original probability values, we have taken the log probabilities and added them.

In section 2.2.3 we have discussed the time complexity for finding the best path (beam) through the lattice. In the test set, there are 9 to 18 words in a sentence. Our module that is implemented in C programming language, takes 9 to 10 seconds for majority of the sentences. It mainly depends on the number of nodes in each level of the lattice that represents the synonym list of the target language words. It is observed that, on an average, the proposed module takes 9.32 seconds per sentence.

3.4 Human Evaluation

The translation accuracy of a machine translation system can be measured using automatic ealuation metric like BLEU score or using human evaluation process. We have tested the translation of 100 sentences through human evaluation and found that 37 sentences are bad translation in both the systems and 16 are equally good. In 13 sentences we found that baseline system is performing better and our system is improving the translation quality in 34 sentences.

4 Conclusion and Future Work

As an initial work we have trained our model on 500K sentence Hindi corpus and tested the performance on Bengali to Hindi transfer based system and shown the improvement in translation quality in terms of BLEU score and Human evaluation. To test the robustness of the proposed scheme, it needs to be tested in other syntactically similar langualge pairs.

We have followed the monotonic search through the lattice and considered a one to one mapping from source to target language. In a transfer based system word reordering, insertion and deletion will be taken care by the transfer grammar features. We have plans to add the transfer grammar features in the lattice and do experiments on reordering in future.

The application of monolingual corpus based lattice structure described here is not only restricted to lexical translation or decoding of target sentence. It may also improve the quality of many other applications of IBM translation models.

ACKNOWLEDGEMENT

This work is partially supported by "The Bangla Treebank" project sponsored by CIIL, Mysore and the ILMT project sponsored by MCIT, Govt of India. We would like to thank all the members of Communication Empowerment Lab, and Mr. Debi Prosad Dogra, CSE, IIT Kharagpur, for their active assistance in the work.

References

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio L. Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Marine Carpuat and Dekai Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72, Prague, Czech Republic, June. Association for Computational Linguistics.

Sanjay Chatterji, Devshri Roy, Sudeshna Sarkar, and Anupam Basu. 2009. A hybrid approach for bengali to hindi machine translation. In *Proceedings of International Conference On Natural Language Processing*, pages 83–91, Hyderabad, India.

Ethnologue. 2011. Ethnologue: Languages of the world, sixteenth edition. Dallas: SIL International. Online version: http://www.ethnologue.com/.

A. Haghighi, P. Liang, T. Berg-Kirkpatrick, and D. Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Human Language Technology and Association for Computational Linguistics (HLT/ACL)*.

Hieu Hoang and Philipp Koehn. 2008. Design of the moses decoder for statistical machine translation. In *SETQA-NLP '08: Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 58–65, Morristown, NJ, USA. Association for Computational Linguistics.

N. Ioannou. 2003. Metis: Statistical machine translation using monolingual corpora. In *In Proceedings of the 6th International Conference of Greek Linguistics*, pages 18–21, University of Crete, Rethymno, Greece.

Mitesh Khapra, Pushpak Bhattacharyya, Shashank Chauhan, Soumya Nair, and Aditya Sharma. 2008. Domain specific iterative word sense disambiguation in a multilingual setting. In *Proceedings of International Conference On Natural Language Processing*, Pune, India.

Hyun Ah Lee, Jong C. Park, and Gil Chang Kim. 1999. Lexical selection with a target language monolingual corpus and an mrd. In *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 99)*, pages 150–160, University College, Chester, England.

Yajuan Lü and Ming Zhou. 2004. Collocation translation acquisition using monolingual corpora. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 167–174, Barcelona, Spain, July.

A. Stolcke. 2002. Srilm – an extensible language modeling toolkit. pages 901–904.

Christoph Tillmann and Hermann Ney. 2003. Word reordering and a dynamic programming beam search algorithm for Statistical Machine Translation. *Computational Linguistics*, 29(1):97–133.

C. Tillmann, S. Vogel, H. Ney, and A. Zubiaga. 1997. A dp based search using monotone alignments in statistical translation. In *Proceedings of 35th Annual Conference of The Association for Computational Linguistics*, pages 289–296.

Appendix A: Itrans Vs Bengali and Hindi unicode Mapping

a অ अ	A আ आ	i ই इ	I ঈ ई	u উ उ	U ঊ ऊ
RRi ঋ ऋ	e এ ए	ai ঐ ऐ	o ও ओ	au ঔ औ	
k ক क	kh খ ख	g গ ग	gh ঘ घ	~N ঙ ङ	
ch চ च	chh ছ छ	j জ ज	jh ঝ झ	~n ঞ ञ	
T ট ट	Th ঠ ठ	D ড ड	Dh ঢ ढ	N ণ ण	
t ত त	th থ थ	d দ द	dh ধ ध	n ন न	
p প प	ph ফ फ	b ব ब	bh ভ भ	m ম म	
y য य	Y য় य़	r র र	l ল ल		
sh শ श	Sh ষ ष	s স स	h হ ह		
.D ড় ड़	.Dh ঢ় ढ़	.n ং ं	H ঃ ः	.N ঁ ँ	